



TRABAJO FIN DE GRADO
GRADO EN INGENIERÍA INFORMÁTICA
MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN

Aplicación móvil para la reeducación de niños con dificultades en el aprendizaje de las matemáticas

Estudiante: Rubén Guede Fernández
Dirección: Paula María Castro Castro
Adriana Dapena Janeiro

A Coruña, febrero de 2020.

A mis padres José y Rosa y a mis hermanos Marta y Pablo.

Agradecimientos

A toda mi familia por estar siempre ahí.

A mis amigos Jorge, Carlos, Lucía y David por todos los momentos compartidos y a todos los que conocí en estos años en la facultad.

A mis tutoras Paula María Castro Castro y Adriana Dapena Janeiro por darme la oportunidad de realizar este proyecto.

Resumen

En este Trabajo Fin de Grado se presenta una aplicación móvil para la plataforma Android orientada a corregir las dificultades en el aprendizaje de las matemáticas a estudiantes de entre cinco y nueve años. La finalidad de esta aplicación es ofrecerles a estos estudiantes una serie de tareas y juegos matemáticos que les ayuden a mejorar su comprensión matemática y la realización de cálculos numéricos. Esta aplicación realizará un seguimiento de la evolución del aprendizaje de un estudiante o de un grupo de ellos. Para ello, otro objetivo que se plantea es la visualización y exportación de esa información. Adicionalmente, se permitirá adaptar la dificultad de estos ejercicios según la evolución de cada estudiante. Por lo tanto, esta herramienta puede constituir un recurso adecuado para una intervención en el aula ante una detección de dificultades de aprendizaje matemático en los primeros años de escolarización.

Abstract

This final degree project presents a mobile application for the Android platform oriented to correct difficulties in learning mathematics for students between five and nine years old. The purpose of this application is to offer the student a series of mathematical tasks and games that help them to improve their understanding and performing numerical calculations. Moreover, this mobile application will keep track of the evolution of a student or group of them, which leads to the additional objective of visualization and exportation of such data. In addition, it is possible to adapt the difficulty of these exercises according to the evolution of each student. Therefore, this tool can be a suitable resource for interventions in case of a detection of difficulties in mathematics learning in the early years of school.

Palabras clave:

- Dificultad
- Aprendizaje
- Matemáticas
- Discalculia
- Móvil

Keywords:

- Difficulty
- Learning
- Math
- Dyscalculia
- Mobile

Índice general

1	Introducción	1
1.1	Motivación	1
1.2	Objetivos	2
1.3	Estructura de la memoria	2
2	¿Qué es la discalculia?	5
2.1	Etiología	5
2.2	Clasificación	6
2.3	Detección	7
2.4	Intervención	7
3	Estado del arte	9
3.1	Smartick	9
3.2	Dinamo Números	10
3.3	Matific	11
3.4	Comparativa entre aplicaciones	11
4	Fundamentos tecnológicos	13
4.1	Gestión del proyecto	13
4.2	Control de versiones	13
4.3	Diseño	14
4.4	Entorno de desarrollo	14
4.5	Lenguajes de desarrollo	14
4.6	Base de datos	14
5	Metodología	15
5.1	Scrum	15
5.2	Scrum aplicado al proyecto desarrollado	18

5.2.1	Product Backlog del proyecto desarrollado	18
5.2.2	Sprints del proyecto desarrollado	19
6	Planificación y estimación de costes	23
6.1	Diagrama de Gantt	23
6.2	Recursos utilizados	23
6.3	Estimación de costes	24
7	Análisis de requisitos	27
7.1	Requisitos funcionales	27
7.1.1	Requisitos funcionales del profesor	27
7.1.2	Requisitos funcionales del alumno	28
7.2	Requisitos no funcionales	28
7.3	Actores	28
7.4	Términos de la aplicación	29
7.5	Historias de usuario	30
7.5.1	Profesor	31
7.5.2	Alumno	33
8	Diseño	37
8.1	Patrones de diseño	37
8.2	Diseño de la base de datos	38
8.3	Shared Preferences	39
8.4	Diseño de la aplicación	39
8.4.1	Profesor	39
8.4.2	Alumno	44
9	Implementación	49
9.1	Implementación	49
10	Pruebas	61
10.1	Pruebas funcionales	61
10.2	Pruebas de usabilidad	63
11	Conclusiones	65
11.1	Líneas futuras de trabajo	66
A	Planificación	71
B	Competencias matemáticas	79

C Contenido del CD adjunto	83
D Manual de usuario. Aplicación móvil	85
D.1 Instalación de la aplicación	85
D.2 Funcionamiento de la aplicación	85
Lista de acrónimos	99
Glosario	101
Bibliografía	103

Índice de figuras

7.1	Diagrama de actores.	29
7.2	Historias de usuario del profesor.	31
7.3	Historias de usuario del alumno.	34
8.1	Modelo de datos.	38
8.2	Diagrama de secuencia de Importar alumnos.	42
8.3	Ejemplo del archivo csv.	43
8.4	Diagrama de secuencia de Exportar estadísticas.	44
8.5	Diagrama de secuencia de Completar ejercicios.	46
A.1	Diagrama de Gantt de los Sprints del proyecto.	72
A.2	Diagrama de Gantt de los Sprints 0 y 1.	73
A.3	Diagrama de Gantt del Sprint 2.	74
A.4	Diagrama de Gantt del Sprint 3.	75
A.5	Diagrama de Gantt del Sprint 4.	76
A.6	Diagramas de las reuniones.	77
A.7	Diagrama de seguimiento al 100%.	78
D.1	Instalación de la aplicación.	86
D.2	Autenticación y registro.	87
D.3	Configurar profesor.	88
D.4	Importar alumnos.	89
D.5	Añadir un alumno.	90
D.6	Configurar Alumno.	91
D.7	Exportar estadísticas de un alumno.	93
D.8	Exportar estadísticas varios alumnos.	94
D.9	Autenticar alumno.	95
D.10	Completar una iteración.	96

D.11 Completar un nivel.	97
D.12 Ver estadísticas.	98

Índice de tablas

3.1	Comparación de las aplicaciones del mercado.	12
6.1	Resumen de horas del proyecto.	25
6.2	Estimación del coste de los recursos humanos.	26
6.3	Estimación del coste total del proyecto.	26
7.1	Prioridades de los casos de uso.	31
11.1	Comparación de la aplicación desarrollada con otras aplicaciones.	66
B.1	Competencias matemáticas de 6º de Educación Infantil.	79
B.2	Competencias matemáticas de 1º de Educación Primaria.	80
B.3	Competencias matemáticas de 2º de Educación Primaria.	81
B.4	Competencias matemáticas de 3º de Educación Primaria.	81

Introducción

EN este capítulo se va a tratar la motivación del trabajo, sus objetivos y la estructura de la memoria de este proyecto.

1.1 Motivación

El bajo rendimiento académico y el abandono escolar de una gran parte del alumnado puede deberse, en muchas ocasiones, a una dificultad en el aprendizaje. De entre todas las dificultades de aprendizaje que existen, en este proyecto nos centramos en la dificultad en el aprendizaje de las matemáticas, conocida como discalculia.

Para comprenderla mejor, podemos describir la discalculia como una dificultad similar a la dislexia, pero que, en lugar de afectar al aprendizaje de la lectura/escritura, afecta al aprendizaje de las matemáticas, a la realización de cálculos, a la comprensión de las relaciones matemáticas, al nombrado de números, símbolos, etc. En el capítulo 2 explicaremos con más detalle estas limitaciones.

Para comprender mejor el alcance de este proyecto, comenzaremos explicando dos términos relacionados con la educación y que se suelen confundir. Por un lado tenemos el **refuerzo educativo**, que consiste en una serie de medidas educativas para reforzar el aprendizaje de las competencias de la materia o materias en las cuales el estudiante presenta más dificultades, aunque estas medidas no pueden orientarse a solventar un desfase curricular superior a los dos años con respecto al curso actual del alumno. El otro término es el de **apoyo educativo**, que es una medida extraordinaria para alumnos con necesidades específicas que no pueden ser atendidas en el refuerzo educativo [1].

En este proyecto se propone el desarrollo de una aplicación móvil que ayude a mejorar las competencias matemáticas de estudiantes de entre cinco y nueve años de edad (ver apéndice B), en los que ya es posible la detección fiable de este tipo de dificultades. Estaríamos hablando de una aplicación orientada a una intervención con un estudiante o grupo de estu-

diantes, fuera de la propia aula de clase, a los que se les ha detectado algún tipo de dificultad de aprendizaje matemático y que, por tanto, necesitan una medida de apoyo educativo a esta necesidad específica.

1.2 Objetivos

El objetivo principal de este proyecto es el desarrollo de una aplicación móvil en el sistema operativo Android que permita realizar una intervención para la mejora de la discalculia con niños del último curso de la etapa de educación infantil (sexto) y primeros cursos de la educación primaria (de primer a tercer curso) [2], mediante la realización y seguimiento de diferentes juegos y ejercicios matemáticos.

El público al que va destinado es un público infantil que presenta discalculia, aunque cualquier estudiante sin este diagnóstico podría usarla ya que se puede ajustar el nivel de dificultad de los ejercicios.

A continuación se presenta un listado de los objetivos que se pretenden conseguir con esta aplicación:

- Desarrollo de una aplicación para móviles y tablets que permita realizar una intervención y ayudar con el aprendizaje de las matemáticas a niños que presentan discalculia.
- Desarrollo de ejercicios y juegos matemáticos con varios niveles de dificultad para los cursos de 6º de educación infantil y 1º, 2º y 3º de educación primaria.
- Implementación de una interfaz gráfica cómoda e intuitiva.
- Implementación de dos perfiles para la aplicación: el profesor, que tendrá el rol de administrador, y el alumno, que tendrá el rol de usuario y será el que realice los ejercicios/juegos matemáticos.
- Visualización y exportación de los resultados de cada alumno, permitiendo realizar un seguimiento de su progresión en el aprendizaje.

1.3 Estructura de la memoria

A continuación se describen brevemente los capítulos en los que se ha estructurado la memoria:

- **Capítulo 1: Introducción.** En este capítulo se hace una breve introducción a la discalculia, a los objetivos que se quieren conseguir con la realización de este Trabajo Fin de Grado y se indica la estructura de esta memoria.

- **Capítulo 2: ¿Qué es la discalculia?** Se aborda el tema sobre el que se basa el proyecto, explicando en qué consiste, cómo detectarla y cómo realizar una intervención.
- **Capítulo 3: Estado del arte.** Se evalúan algunas herramientas del mercado que tienen relación con el actual proyecto.
- **Capítulo 4: Fundamentos tecnológicos.** Se describen las herramientas usadas para la realización de este proyecto.
- **Capítulo 5: Metodología.** En este capítulo se habla de la metodología ágil utilizada en el desarrollo del proyecto, sus características y los elementos que la componen.
- **Capítulo 6: Planificación y estimación de costes.** Se explica la planificación inicial y final del proyecto y la estimación del coste del mismo.
- **Capítulo 7: Análisis de requisitos.** Se describen los requisitos de la aplicación y los actores que interactúan con ella.
- **Capítulo 8: Diseño.** Se describen los aspectos relacionados con el diseño de la aplicación.
- **Capítulo 9: Implementación:** Se detallan aspectos relevantes de la implementación del proyecto.
- **Capítulo 10: Pruebas:** Se comentan las pruebas realizadas de los casos de uso de la aplicación.
- **Capítulo 11: Conclusiones:** Se comprueba el cumplimiento de los objetivos iniciales, se resaltan los aspectos principales del trabajo realizando y se exponen funcionalidades que se podrían añadir o mejorar sobre la aplicación desarrollada.

¿Qué es la discalculia?

EL término discalculia es una palabra griega, compuesta por dis («dificultad con») y calculia («cálculos medios»). Se trata de una dificultad de aprendizaje poco conocida, por lo que no se suele tratar adecuadamente. Este trastorno dificulta la realización de cálculos matemáticos básicos, el conteo, la definición de grupos de objetos y el pensamiento espacial [3].

En 1974, L. Kosc la definió como «dificultad en el funcionamiento matemático resultado de un trastorno del procesamiento matemático de origen cerebral sin compromiso de otras áreas del aprendizaje» [3]. Esta definición se sigue usando a día de hoy. Algunos autores la consideran, sin embargo, como un tipo de dislexia, porque los que sufren esta dificultad suelen presentar dificultades en el aprendizaje de la lectura y la escritura.

Al igual que la dislexia, la discalculia puede afectar a personas con una inteligencia normal o superior a la media. También pueden aparecer problemas psicológicos derivados como una baja autoestima, problemas sociales, emocionales, etc.

2.1 Etiología

Existen diferentes enfoques que intentan explicar el origen o las causas de las dificultades en el aprendizaje de las matemáticas. Estos enfoques son los siguientes:

- **Enfoque evolutivo:** Pone énfasis en la estimulación recibida por el niño en su desarrollo desde que se manifiesta la discalculia.
- **Enfoque educativo:** Se centra en la dificultad de la asignatura y en su enseñanza, destacando las estrategias utilizadas por los profesores para enseñar a sus alumnos.
- **Enfoque neurológico:** Relaciona las lesiones cerebrales con las dificultades en el aprendizaje de las matemáticas. En el hemisferio derecho encontramos las bases de las principales funciones matemáticas, como el alineamiento de números o la resolución de

problemas. El hemisferio izquierdo es el responsable de la comprensión y expresión del lenguaje, como puede ser leer y escribir números.

- **Enfoque cognitivo:** Este enfoque establece que las dificultades se deben a procesos cognitivos erróneos o inadecuados utilizados para resolver el problema matemático. Se deben identificar los procesos mentales usados para la resolución del problema y cambiarlos por otros que consigan llegar a la solución del problema matemático.

Al igual que ocurre con la dislexia, el origen exacto de la discalculia es incierto. Se sabe que afecta a varias áreas del desarrollo, como a la capacidad visoespacial y visiooperceptiva.

Sin una correcta intervención, el niño o niña con discalculia va arrastrando dificultades matemáticas a lo largo de su desarrollo académico, lo que podría desembocar en un fracaso escolar.

2.2 Clasificación

En 1926, Berger hizo una separación entre discalculia primaria y secundaria. Este autor define la discalculia primaria como el trastorno del cálculo que produce una lesión cerebral que no incluye alteraciones del lenguaje o del razonamiento. La secundaria, en cambio, la atribuye a alteraciones verbales, espaciotemporales o de razonamiento [3].

En la década de los sesenta, Hécaen y su equipo hicieron una clasificación de los trastornos del cálculo producidos por lesión cerebral [3]. Distinguen tres tipos:

- **Alexia y agrafía numérica:** Alteraciones en la lectura y escritura de los números.
- **Acalculia espacial:** Alteraciones en la colocación en el espacio de los números.
- **Anaritmética:** Incapacidad primaria del cálculo. Se corresponde con la discalculia primaria de Berger.

En 1974, Kosc clasificó el trastorno asociado a las dificultades del aprendizaje de las matemáticas en seis tipos: [3]

- **Verbal:** Incapacidad para entender conceptos matemáticos y relaciones presentadas oralmente.
- **Practognóstica:** Trastorno en la manipulación del tamaño de los objetos.
- **Lexical:** Dificultad para tratar los símbolos matemáticos, por ejemplo, confundir la suma con la multiplicación.
- **Gráfica:** Dificultad para escribir símbolos matemáticos.

- **Ideognóstica:** Dificultad para entender conceptos matemáticos y para hacer cálculos mentales.
- **Operacional:** Dificultad para realizar operaciones matemáticas.

2.3 Detección

Las dificultades en el aprendizaje de las matemáticas necesitan en muchas ocasiones de un tratamiento paralelo de otras dificultades con las que presenta interacciones, como la disortografía (dificultades en la asociación entre el código escrito, las normas ortográficas y la escritura de palabras), disgrafía (dificultad para coordinar los músculos de la mano y del brazo, lo que impide dominar y dirigir el lápiz para escribir de forma legible y ordenada) o la dislexia (dificultad en el aprendizaje de la lectura y escritura).

La detección es esencial para llevar a cabo una correcta intervención, y en la discalculia es fundamental que se produzca lo antes posible. El primer paso para la detección es descartar posibles anomalías de origen neurológico.

Las dificultades que presentan los alumnos de estas primeras etapas educativas suelen ser:

- Reconocimiento y comprensión de números.
- Problemas de memoria que dificultan recordar los números en el orden correcto.
- Dificultad para entender conceptos como el conjunto (grande, pequeño).
- Problemas para entender los conceptos de medida.

2.4 Intervención

Normalmente, las dificultades en el aprendizaje de las matemáticas se descubren en el ámbito escolar. Según Piaget [3], la comprensión de operaciones aritméticas como la adición y la sustracción no se alcanza hasta los 5 años. Además, hasta los 6 ó 7 años, no se entiende el concepto de cantidad. Por lo tanto, no podemos decir que el niño puede presentar discalculia hasta que no haya alcanzado una edad de entre 5 y 9 años.

Para una correcta intervención, además de una evaluación del alumno, es necesario realizar entrevistas con la familia, para obtener información del entorno familiar, y con su tutor, para saber más sobre su entorno escolar.

En la entrevista con el tutor es necesario preguntar sobre la situación familiar del alumno, por ejemplo, si ha sufrido la pérdida de algún familiar, o si sus padres se han separado, en definitiva, si ha habido cambios importantes en su núcleo familiar, ya que esto puede afectar al rendimiento escolar del niño. Además, se debe obtener información de las asignaturas en

las que presenta mayores dificultades, qué tipo de dificultades aparecen y si el alumno está recibiendo refuerzo educativo en alguna de ellas.

La entrevista con la familia, si es posible, debe hacerse en la residencia del alumno para obtener así información relevante de su entorno, como puede ser la relación entre los progenitores o tutores del niño, si éste utiliza o no una habitación propia para estudiar, el número de hermanos, si los hay, y sus relaciones con ellos, es decir, para conocer, en general, las condiciones en las que se desarrolla su vida diaria. Es importante saber en qué momento los padres se dieron cuenta de que su hijo tenía problemas con las matemáticas, ya que, aunque los niños con discalculia presentan un bajo rendimiento desde los primeros cursos de primaria, éstos pueden presentarse ya en los primeros años de la infancia. También es interesante conocer si en la familia hubo antecedentes con algún tipo de dificultad en el aprendizaje como, por ejemplo, la dislexia.

En cuanto a la evaluación del alumno, se llevan a cabo una serie de ejercicios de evaluación orientados básicamente a:

- Determinar el nivel de ejecución aritmética.
- Resolver problemas sencillos aplicando la suma, la resta, la multiplicación y la división con números naturales.
- Leer, escribir y ordenar números sencillos enteros y decimales.
- Reconocer y describir formas y cuerpos geométricos.

En la evaluación también se tienen en cuenta aspectos personales como la motivación, el estado anímico, o las relaciones personales.

Una vez que existe la sospecha de que el alumno puede presentar discalculia, debemos someterle a una serie de pruebas estandarizadas para corroborar esa sospecha y determinar su alcance, es decir, el grado de dificultad. Entre las pruebas más utilizadas destacan:

- **Pruebas de inteligencia (CI):** Permiten saber el coeficiente intelectual de alumno y así saber si su dificultad se deriva de su CI. El Test de inteligencia General Factorial (IGF-6R) [4] y las Pruebas de Inteligencia Wechsler (WISC-R [5]), WISC-IV [6]) son algunas de las pruebas que se realizan para determinar su CI.
- **Pruebas de competencia curricular:** Para que se considere que existe un desfase curricular éste ha de ser inferior a dos cursos escolares con respecto al correspondiente a la edad del alumno. Destacamos algunas pruebas como, el Test para el diagnóstico de las Competencias Básicas en matemáticas (TEDI-MATH) [7], o la Batería de Aptitudes Diferenciales y Generales (BADYG) [8].

Estado del arte

EN este capítulo se presentará una revisión de las aplicaciones relacionadas con el tema principal de este trabajo. Comentaremos qué características presentan estas herramientas, el tipo de público al que va destinado y destacaremos sus puntos fuertes y débiles.

3.1 Smartick

Es una “Startup” fundada en 2011 por dos emprendedores españoles -Daniel González de Vega y Javier Arroyo [9]. Utiliza Inteligencia Artificial y Big Data para la generación de ejercicios, permitiendo que cada alumno tenga un plan de ejercicios personalizado y adaptable a su evolución.

Es una herramienta de aprendizaje de las matemáticas para niños de 4 a 14 años. Está disponible actualmente para PCs, tablets y móviles (IOS y Android). De entre todas las herramientas encontradas es la que más similitudes tiene con nuestro proyecto, ya que trata dificultades como la discalculia, entre otras muchas y está disponible para móviles y tablets con sistema operativo Android.

Nos centraremos en su versión para dispositivos móviles y tablets, ya que el objetivo de la aplicación que se va a desarrollar es que funcione correctamente en estos dispositivos. Smartick tiene auto-aprendizaje, es decir, permite que el alumno avance sin necesidad de un profesor que lo dirija. Cuando aparece un concepto que el alumno no entiende la herramienta le indica de forma visual cómo resolver el ejercicio mostrándole su resolución paso a paso. Estas ayudas gráficas también aparecen cuando la herramienta detecta que el alumno tarda mucho en responder.

Cuenta con una gran variedad de ejercicios, entre los que destacan los ejercicios de cálculo, geometría, razonamiento y lógica. La aplicación le permite al alumno tomar algunos descansos como si fueran recreos y, al volver, seguir en el ejercicio en el punto en el que lo dejó. Además, cuenta con corrección automática, de forma que al finalizar cada ejercicio se le muestra al

alumno la solución del mismo.

Otro de sus puntos fuertes es la cantidad de dificultades de aprendizaje que contempla: discalculia [10], discapacidad intelectual [10], discapacidad motora [11], discapacidad sensorial auditiva [10], dislexia [10], parálisis cerebral [11], síndrome de Down [12], Asperger [13], Trastorno de Espectro Autista (TEA) [10] o Trastorno por Déficit de Atención e Hiperactividad [10] (TDAH) [10].

Como punto débil, conviene destacar que no cuenta con un seguimiento de la evolución del alumno, característica que sí comparten las otras aplicaciones encontradas.

3.2 Dinamo Números

Dinamo Números es una herramienta web que pertenece a la editorial JellyJames Publishing Ltd [14]. Su objetivo principal es la evaluación e intervención estándar de la discalculia. Su programa se basa en cuatro etapas que siguen un proceso cíclico: evaluar, planear, intervenir y revisar.

Esta primera fase de evaluación de la discalculia detecta si el alumno presenta la discalculia o si va a desarrollarla en un futuro. Además muestra los puntos fuertes y débiles de cada alumno. El aspecto clave de esta evaluación reside en la diferenciación entre los alumnos que han desarrollado discalculia y los que sufren retrasos en el aprendizaje de las matemáticas, ya sea por problemas familiares o por una educación mediocre.

La siguiente fase, la de planificación, permite obtener un plan personalizado para cada alumno, utilizando los resultados obtenidos en la fase anterior.

La fase de intervención sigue un enfoque en tres pasos: una programación de aula, actividades online y hojas de trabajo. Ofrece múltiples formas de aprender una misma habilidad, por ejemplo, utilizando elementos visuales, auditivos y las habilidades kinestésicas del estudiante.

La última fase, la de revisión, permite visualizar los resultados de cada alumno, ya sea mediante tablas, diagramas o gráficas, lo que permite realizar un seguimiento de sus avances a lo largo del tiempo.

Por último, destacar que Dinamo Números ha sido nominada a varios premios, con los siguientes resultados:

- Finalista del premio BETT (2017).
- Finalista del premio ERA (2016).
- Finalista del premio BETT (2015).
- Finalista del premio BETT (2014).

Estos premios BETT [15] son parte de una feria anual que se centra en las tecnologías de la información dentro del ámbito de la educación.

Los premios ERA [16] son un evento anual en el que destacan y elogian la calidad de los productos educativos, de recursos y del personal docente, y es reconocido como un premio a la excelencia.

3.3 Matific

Es una herramienta que ayuda con las matemáticas a estudiantes desde educación infantil hasta primaria [17]. Se encuentra disponible para PCs, móviles, tablets e incorpora más de 40 idiomas.

Cuenta con un algoritmo inteligente de asignación automática para saber qué tareas asignar a cada alumno. Permite crear rutas de aprendizaje diferenciadas para que los estudiantes progresen a su propio ritmo.

En cuanto a los niveles, Matific cuenta con episodios, que fomentan la exploración y la resolución de problemas, mediante niveles interactivos, diseñados para lograr el autoaprendizaje del alumno. Otro tipo de niveles son las hojas de trabajo electrónicas, que tratan operaciones como la suma, la resta, la multiplicación o la división. El tercer tipo de niveles son los problemas textuales o tareas que tratan situaciones reales y que buscan el desarrollo del pensamiento crítico del alumno. Por último, tenemos los talleres, un entorno interactivo donde los estudiantes aprenden conceptos como la cantidad, las fracciones o la resta.

La aplicación proporciona informes de seguimiento del progreso del alumno. Estos informes son automatizados y en tiempo real, permitiendo al profesor seguir el progreso de sus alumnos.

3.4 Comparativa entre aplicaciones

En este apartado se muestra una tabla comparativa de las funcionalidades incorporadas en cada una de las aplicaciones anteriores. Se han seleccionado cinco características que se consideran importantes en una aplicación que trate la discalculia. Como puede observarse, la aplicación Smartick cumple con todas las características y se considera la más completa de las tres. A la aplicación Dinamo Números le faltaría estar disponible en aplicaciones móviles y, en cuanto a matific, no se centra en tratar la discalculia, sino en ofrecer un conjunto de actividades que sirven como un complemento adicional al programa escolar utilizado por un profesor.

La aplicación que se va a desarrollar pretende incorporar la mayoría de las funcionalidades de todas estas aplicaciones, potenciando sus puntos fuertes y solventando las carencias

3.4. COMPARATIVA ENTRE APLICACIONES

	Smartick	Dínamo Números	Matific
Más de un tipo de dispositivo (móvil, tablet, web)	✓	X	✓
Auto-aprendizaje	✓	✓	✓
Variedad de ejercicios	✓	✓	✓
Controlar el seguimiento por parte de un profesor o tutor	✓	✓	✓
Tiene como objetivo ayudar a mejorar la discalculia	✓	✓	X
Exporta información del seguimiento del estudiante	X	✓	X

Tabla 3.1: Comparación de las aplicaciones del mercado.

detectadas.

Fundamentos tecnológicos

EN este capítulo se van a tratar las herramientas que se han usado a lo largo del desarrollo del proyecto.

4.1 Gestión del proyecto

Microsoft Project

Microsoft Project [18] es un software que permite gestionar un proyecto. Fue comercializado por Microsoft para ayudar a los responsables de los proyectos a planificar las tareas, asignar los recursos a las mismas, realizar un seguimiento del proyecto y administrar el presupuesto.

Por todo esto y estar familiarizado con él tras usarlo previamente, se ha elegido este software para realizar la planificación, el seguimiento y la estimación de los costes del actual proyecto.

4.2 Control de versiones

Git y GitHub

Se ha utilizado el sistema de control de versiones Git [19]. Como plataforma para alojar el proyecto se ha usado GitHub [20]. Es una de las plataformas más extendidas en lo que se refiere a desarrollo colaborativo, permitiendo trabajar desde diferentes equipos y disponer de una copia de seguridad de todas las versiones. Permite alojar proyectos de forma pública y desde el año 2019 permite también alojarlos de manera privada con la cuenta gratuita.

4.3 Diseño

Se ha utilizado el programa Dia [21] junto con la herramienta Visual Paradigm Online [22] para realizar los diagramas de secuencia y el diseño de la base de datos que se muestran en el presente documento.

4.4 Entorno de desarrollo

Android Studio

Como entorno de desarrollo se ha escogido Android Studio [23]. Es el entorno de desarrollo oficial para el sistema operativo Android y está basado en software IntelliJ IDEA de JetBrains. El sistema de compilación se basa en gradle, un complemento que cuenta con características específicas para compilar aplicaciones en Android. Además, este entorno cuenta con un emulador, que permite realizar pruebas en dispositivos de tamaños diferentes como móviles y tablets.

4.5 Lenguajes de desarrollo

Se ha utilizado Java como lenguaje de programación para la implementación de los casos de uso y los modelos, Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems [24]. El lenguaje xml [25] se ha usado para la implementación de la parte gráfica de la aplicación.

4.6 Base de datos

SQLite

Como sistema de gestión de bases de datos se ha usado SQLite [26]. Se ha escogido porque permite realizar transacciones ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad). Se encuentra integrado en todos los teléfonos móviles, es de dominio público y permite la retrocompatibilidad de sus datos.

Capítulo 5

Metodología

UNA metodología en desarrollo software es un marco de trabajo que nos permite planificar, gestionar y administrar un proyecto, permitiendo obtener un producto cercano al esperado. Existen dos tipos de metodologías, las tradicionales y las ágiles. Las primeras son más rígidas, ya que no se adaptan correctamente a los cambios. Sin embargo, en las metodologías ágiles se obtienen versiones funcionales del producto sin tener que esperar a que finalice el proyecto y se adaptan ante los posibles cambios que puedan ir surgiendo [27].

Para este proyecto se ha optado por Scrum, una metodología ágil de ciclo de vida iterativo e incremental. Existen múltiples herramientas para implementar este tipo de metodologías: Atlassian JIRA [28], VersionOne [29] o Trello [30], entre otras.

Destacamos una serie de beneficios de estas metodologías ágiles frente a las tradicionales:

- Respuesta rápida ante los cambios.
- Obtención de software funcional al final de cada iteración.
- Más calidad en el producto final.
- Mayor participación del cliente.

5.1 Scrum

En este apartado se explicará en qué consiste la metodología ágil Scrum que se ha utilizado en este proyecto.

Elementos

El marco de trabajo Scrum se basa en los Equipos Scrum, eventos, artefactos y reglas asociadas [31].

El Equipo Scrum está formado por tres roles. Cada rol describe la función que realizará cada miembro del equipo. Distinguimos entre el Dueño de Producto (Product Owner), el Equipo de Desarrollo (Development Team) y el Scrum Master.

Scrum utiliza iteraciones llamadas Sprints, que abarcan un marco temporal en el que llevar a cabo las tareas y tienen una duración máxima de un mes.

Artefactos

A continuación describimos los distintos artefactos que se consideran en esta metodología.

Lista de Producto (Product Backlog): Contiene todas las características, funcionalidades, requisitos, mejoras y correcciones que tendrá el producto final.

El product backlog nunca está completo, ya que va evolucionando a medida que se va avanzando con el producto.

Lista de Pendientes del Sprint (Sprint Backlog): Contiene los elementos del product backlog que se van a realizar en el Sprint actual y un plan detallado para lograr el incremento del producto, consiguiendo así el objetivo del Sprint.

Incremento: Es la suma de todos los elementos de la product backlog completados durante un Sprint y el valor de los incrementos de todos los Sprints anteriores.

Sprints

Representan el núcleo de Scrum. La idea es que en cada Sprint se lleve a cabo el ciclo de desarrollo software: requisitos, análisis, desarrollo, codificación y pruebas. Se busca conseguir un producto entregable en cada Sprint.

Scrum considera cuatro eventos que se llevan a cabo en cada Sprint, cada uno con una duración máxima. La duración máxima de cada evento es una estimación, lo que quiere decir que una vez que se ha alcanzado el objetivo de ese evento se pasa al siguiente. Los eventos son: Planificación del Sprint (Sprint Planning), Scrum Diario (Daily Scrum), Revisión del Sprint (Sprint Review) y Retrospectiva del Sprint (Sprint Retrospective). En todos los eventos el Scrum Master se asegura de que no se supera el tiempo fijado y de que cada persona entiende el propósito del evento. Detallamos a continuación cada uno de estos eventos.

Planificación del Sprint (Sprint Planning): Se planifica el trabajo que se va a hacer durante el Sprint. Tienen una duración máxima de ocho horas para un Sprint de un mes. Como se ha comentado antes, una vez que se ha planificado con éxito el Sprint, ya se ha alcanzado el objetivo de este evento, así que se da por finalizado aunque la duración de la planificación durase dos horas.

El dueño del producto argumenta qué funcionalidades del product backlog se deben llevar a cabo. El número de elementos del product backlog que se seleccionarán depende del equipo de desarrollo. Solo ellos saben lo que son capaces de conseguir en el Sprint.

Scrum Diario (Daily Scrum): Es una reunión de duración máxima de 15 minutos con el objetivo de solucionar las posibles dudas que existan con el desarrollo normal del Sprint. El Scrum Diario se realiza a la misma hora y en el mismo lugar todos los días. Mejora la comunicación y se identifican posibles bloqueos o dudas que tenga el equipo de desarrollo. Scrum recomienda que durante esta reunión cada miembro del equipo de desarrollo responda a tres preguntas, para lograr el objetivo de este evento que indicábamos antes:

- ¿Qué hice ayer que ayudó al equipo de desarrollo a lograr el objetivo del Sprint?
- ¿Qué haré hoy para ayudar al equipo de desarrollo a lograr el objetivo del Sprint?
- ¿Qué bloques o dudas tengo que eviten que el equipo de desarrollo logre el objetivo del Sprint?

Revisión del Sprint (Sprint Review): Es una reunión informal que se lleva a cabo al finalizar cada Sprint. Tiene una duración máxima de cuatro horas para Sprints de un mes de duración. Participan los componentes del equipo Scrum y exponen lo que se ha conseguido durante el Sprint.

Retrospectiva del Sprint (Sprint Retrospective): Se lleva a cabo después de la revisión del Sprint y antes de la planificación del siguiente Sprint. Tiene una duración máxima de tres horas para Sprints de un mes de duración. Los objetivos que se busca conseguir con la retrospectiva son:

- Saber cómo fue el Sprint en cuanto a personas, relaciones y procesos.
- Identificar los aspectos que salieron bien y las posibles mejoras.
- Crear un plan para implementar las mejoras en el siguiente Sprint.

Roles

Describimos ahora los distintos roles contemplados en esta metodología:

El Dueño del Producto (Product Owner): Es una única persona y es responsable de maximizar el valor del producto y el trabajo del Equipo de Desarrollo. También se encarga de gestionar el product backlog.

Se tienen que respetar sus peticiones, nadie puede pedirle al equipo de desarrollo que trabaje con requisitos distintos a los indicados en el product backlog y el equipo de desarrollo tampoco puede hacer caso a lo que diga cualquier otra persona que no sea el dueño del producto.

Equipo de Desarrollo (Development Team): Representa a las personas que se encargan de desarrollar las tareas del product backlog. Se busca que el equipo de desarrollo tenga las siguientes características:

- Equipo autoorganizado.
- Equipo multifuncional.
- Todos los miembros son considerados como desarrolladores, sin títulos individuales.
- Pueden existir miembros con habilidades en áreas específicas, pero la responsabilidad recae en el grupo como un todo.

Scrum Master: Es el responsable de que se entienda y se adopte Scrum. Se encarga de que el equipo siga las reglas indicadas en Scrum.

5.2 Scrum aplicado al proyecto desarrollado

Debido a que estamos ante un Trabajo Fin de Grado, no es posible seguir todas las reglas que nos recomienda Scrum. Por ello, se ha adaptado esta metodología a los recursos humanos y necesidades del proyecto. Se han distribuido los roles de la siguiente manera:

- Product Owner: Este rol es llevado a cabo por Paula María Castro Castro, una de las codirectoras de este proyecto.
- Scrum Master: Este rol es realizado por Adriana Dapena Janeiro, otra de las codirectoras de este proyecto.
- Development Team: Este es uno de los aspectos que se han adaptado, ya que Scrum recomienda equipos de desarrollo de entre cinco y nueve personas. En este caso estará formado por el alumno Rubén Guede Fernández.

Como se ha mencionado en el apartado 5.1, en cada uno de ellos se llevan a cabo todas las fases del ciclo de vida de un producto. Además, al final de cada Sprint se ha entregado un documento que posteriormente ha ayudado a realizar la memoria de una forma más ágil. Para una mejor comprensión del trabajo realizado, se ha decidido realizar una presentación de las tareas realizadas en cada Sprint y separar las diferentes fases del ciclo de vida en capítulos independientes.

5.2.1 Product Backlog del proyecto desarrollado

En esta parte se enumerarán las funcionalidades de los distintos productos intermedios que forman el presente proyecto:

0. Investigación.

- Búsqueda de información sobre la discalculia.
 - Revisión y búsqueda de tecnologías y herramientas para el desarrollo del proyecto.
1. Desarrollo de ejercicios matemáticos.
 - Implementación de un conjunto de ejercicios matemáticos.
 - Resultados de los ejercicios completados.
 - Seguimientos de los ejercicios.
 2. Gestión de usuarios.
 - Alta en el sistema de un usuario de tipo “Profesor”.
 - Alta en el sistema de un usuario de tipo “Alumno”.
 - Autenticación de un profesor en el sistema.
 - Autenticación de un alumno en el sistema.
 - Cerrar la sesión.
 - Eliminación de la cuenta de un usuario.
 - Datos personales del usuario de tipo “Profesor”.
 - Datos personales del usuario de tipo “Alumno”.
 - Listado de alumnos.
 - Listado de estadísticas.
 3. Desarrollo de un mecanismo de importación y exportación de datos.
 - Importación de alumnos en el sistema.
 - Exportación de las estadísticas de los alumnos.
 4. Gestión de correos y finalización de la memoria.
 - Correo de bienvenida y de recuperación de contraseña.
 - Memoria.

5.2.2 Sprints del proyecto desarrollado

En este apartado se van a explicar los Sprints en los que se divide el proyecto y qué casos de uso tiene cada uno de ellos.

Se ha marcado como inicio de proyecto el 22 de Noviembre de 2019 y con finalización el 6 de Febrero de 2020. La planificación inicial se ha hecho utilizando Sprints, tal y como indica la metodología Scrum. No todos los Sprints tienen la misma duración, ya que ésta se ha estimado

en función de las tareas a desarrollar en cada uno y de su dificultad. Conviene destacar que la elaboración de la memoria no pertenece a un Sprint concreto, sino que se ha ido elaborando a lo largo de todos los Sprints.

Sprint 0: Investigación

Este Sprint representa la fase inicial del proyecto. Tiene una duración de una semana y se ha utilizado para buscar información sobre la discalculia, las herramientas software y los lenguajes de programación que serán más útiles para el proyecto, con el objetivo de realizar una estimación de la duración y alcance del proyecto lo más realista posible.

- Búsqueda de información sobre la discalculia: Se ha recopilado información sobre la discalculia, qué tipos existen, cómo se diagnostica y qué métodos de intervención son más adecuados.
- Búsqueda de tecnologías y herramientas para el desarrollo del proyecto: Es importante tener claro desde el principio qué herramientas y tecnologías se van a usar y las dificultades que pueden presentar, todo ello para contribuir a llevar a cabo una correcta planificación.

Al finalizar este Sprint se sabe qué tecnologías y herramientas se van a usar para el desarrollo del proyecto. Además se obtiene una estimación inicial del tiempo necesario para su realización y del coste total del proyecto.

Sprint 1: Desarrollo de ejercicios matemáticos

Este Sprint es el núcleo de la aplicación ya que comprende el diseño, implementación y pruebas de los distintos niveles con los que el usuario puede interactuar. Por lo tanto, es el sprint al que más duración se le ha asignado. Tiene una duración de un mes.

- Implementación de un conjunto de ejercicios y juegos matemáticos: Crear y mostrar una lista de ejercicios al alumno.
- Resultados de ejercicios completados: Mostrar los resultados de los ejercicios completados por el alumno.
- Seguimiento de los ejercicios: Llevar a cabo en mecanismo que permita realizar un seguimiento del progreso del alumno.

Hito 1: Primera versión de la aplicación.- Se obtiene una aplicación que tiene implementados todos los ejercicios, es posible ver los resultados y se puede llevar a cabo un seguimiento de los ejercicios.

Sprint 2: Gestión de usuarios

Este Sprint tiene una duración de tres semanas y se llevan a cabo los casos de uso relacionados con la gestión de los usuarios y de sus estadísticas.

- Alta en el sistema de un usuario de tipo “Profesor”: Implementar un mecanismo para registrar profesores en la aplicación.
- Alta en el sistema de un usuario de tipo “Alumno”: Implementar un mecanismo para añadir alumnos en la aplicación.
- Autenticación de un profesor en el sistema: Permitir iniciar sesión a un profesor en la aplicación.
- Autenticación de un alumno en el sistema: Permitir iniciar sesión a un alumno en la aplicación.
- Cierre de la sesión: Habilitar un mecanismo para que el usuario pueda cerrar su sesión activa.
- Eliminación de la cuenta de un usuario: Implementar un mecanismo para eliminar del sistema los datos de un usuario.
- Datos personales del usuario de tipo “Profesor”: Mostrarle al profesor sus datos y permitirle modificar sus datos.
- Datos personales del usuario de tipo “Alumno”: Mostrarle al alumno sus datos y permitirle modificar sus datos.
- Listado de alumnos: Mostrar una lista de los alumnos que tiene un profesor.
- Listado de estadísticas: Mostrar una lista de las estadísticas que tiene un alumno.

Hito 2: Gestión de usuarios en la aplicación.- Partiendo de la aplicación obtenida en el anterior Sprint, se permite registrar y eliminar usuarios y los usuarios pueden ver y modificar sus datos personales. En cuanto a las estadísticas, los alumnos pueden ver sus resultados conseguidos al finalizar los ejercicios. Los profesores también pueden consultar las estadísticas de sus alumnos, así como la visualización y modificación de los datos personales de estos.

Sprint 3: Desarrollo de un mecanismo de importación y exportación de datos

Esta iteración tiene una duración de una semana, en la que se llevan a cabo los casos de uso de importar alumnos y exportar sus estadísticas.

- Importación de alumnos en el sistema: Implementar un mecanismo para permitir añadir múltiples usuarios a través de un archivo en formato csv.
- Exportación de las estadísticas de los alumnos: Implementar un mecanismo que permita la exportación de las estadísticas de uno o varios alumnos por parte del profesor.

Hito 3: Importación y exportación en la aplicación.- Partiendo de la aplicación obtenida en el anterior Sprint, se le permite al profesor importar varios alumnos a través de un archivo csv y exportar las estadísticas de los alumnos seleccionados fuera de la aplicación.

Sprint 4: Gestión de correos y finalización de la memoria

Este Sprint tiene una duración de una semana. Se llevan a cabo las tareas de generar correos y finalización de la memoria.

- Generación de correos: Generar y enviar un correo de bienvenida y un correo de recuperación de contraseña de los profesores.
- Memoria: Completar la documentación de la memoria.

Hito 4: Correos en la aplicación y documentación de la memoria. Se finaliza la elaboración del actual documento y se consigue una versión final de la aplicación con todas las funcionalidades implementadas, probadas y cumpliendo con los objetivos marcados al inicio del proyecto.

Planificación y estimación de costes

EN este capítulo se va a mostrar la planificación inicial que se ha estimado para la realización de este proyecto, junto con un cálculo aproximado del coste del mismo.

6.1 Diagrama de Gantt

Planificación inicial

En la figura [A.1](#) se puede ver el diagrama de Gantt con la planificación inicial del proyecto. En este apéndice se incluyen también las figuras [A.2](#), [A.3](#), [A.4](#) y [A.5](#) en las que se puede ver la planificación de los distintos Sprints que componen el proyecto y detallados en el capítulo anterior. Por último, en la figura [A.6](#) se puede observar cómo se han distribuido las reuniones en las que estuvieron presentes los miembros del equipo que forman este proyecto.

Seguimiento al 100%

En la figura [6.1](#) se puede ver el seguimiento una vez acabado el proyecto. Se puede observar cómo el proyecto ha aumentado su duración en cuatro días, debido a que la realización de la memoria en el último Sprint llevó más tiempo de lo estimado. El resto de plazos establecidos en la planificación inicial han sido cumplidos con éxito.

6.2 Recursos utilizados

Durante el desarrollo de este proyecto se han usado los siguientes recursos hardware y software:

- Hardware:
 - Ordenador portátil:

- * Modelo: Asus GL552VW
- * CPU: i7-6700HQ a 2.6GHz
- * Disco duro: 1TB 7200rpm SATA
- * Memoria RAM: 8GB
- * SO: Windows 10 Home 64 bit
- * Tarjeta gráfica: NVIDIA GeForce 960M DE 2 GB
- Móvil:
 - * Dispositivo: Samsung Galaxy S7
 - * Modelo: SM-G930F
 - * Memoria RAM: 4GB
 - * SO: Android
- Software:
 - Android Studio
 - GitHub
 - Microsoft Project
 - Dia
 - Visual Paradigm Online

Estos recursos software se encuentran definidos en el capítulo 4.

6.3 Estimación de costes

En este apartado se realizará una estimación total del coste asociado a la realización de este proyecto. Desglosaremos este coste en tres partes: hardware, software y equipo humano.

- Costes estimados del hardware: El coste del hardware se calcula como el tiempo de uso de este tipo de recursos durante el proyecto (3 meses) en relación con su tiempo de vida. Para un portátil tenemos en cuenta que su vida media son cuatro años, por lo que imputándole una estimación de 800 €, tenemos que $800 \text{ €} / 4 \text{ años} \times 1/4 = 50 \text{ €}$. Teniendo en cuenta que la vida media de un móvil es de 2 años, imputándole un coste de 300 €, su coste estimado en el proyecto será de $300 \text{ €} / 2 \text{ años} \times 1/4 = 37,50 \text{ €}$. Por lo tanto, tenemos que el coste del hardware es de 87,50 €.
- Costes estimados del software:
 - Android Studio: Es una herramienta gratuita por lo que su coste es nulo.

- GitHub: Se ha usado un repositorio privado para alojar el proyecto. El repositorio privado es gratuito, por lo que su coste también es nulo.
- Microsoft Project: Es una herramienta de pago, que tiene un coste de 25,30 €/mes. En tres meses, el coste estimado será de 75,90€.
- Dia: Esta herramienta utilizada para realizar los diagramas es de código abierto, por lo que su coste es nulo.
- Visual Paradigm Online: Al igual que la herramienta anterior, su coste es nulo.

Por lo tanto, el coste total estimado del software que se necesita para la realización de este proyecto es de 75,90€.

- Coste estimado de recursos humanos: Se distinguen dos categorías: analista programador, categoría desempeñada por el alumno Rubén Guede Fernández, y directora de proyecto, que lo realizan las profesoras Adriana Dapena Janeiro y Paula María Castro Castro.

En la tabla 6.1 podemos ver las horas realizadas en cada Sprint por el analista programador.

Sprint	Objetivo	Fecha Inicio	Fecha Fin	Duración (horas)
0	Investigación	22/11/2019	26/11/2019	20
1	Desarrollo de ejercicios	27/11/2019	24/12/2019	155
2	Gestión de usuarios	02/01/2020	21/01/2020	130
3	Importación y exportación de datos	22/01/2020	29/01/2020	42
4	Gestión de correos y finalización de la memoria	30/01/2020	10/02/2020	84

Tabla 6.1: Resumen de horas del proyecto.

Las jornadas de trabajo han sido de aproximadamente 7 horas al día, a lo que hay que sumar el tiempo empleado en las reuniones del equipo que tuvieron aproximadamente una duración de hora y media. En total se han estimado 431 horas/persona para el analista programador, de acuerdo con la tabla 6.1, y 21 horas/persona la directora de proyecto. Teniendo en cuenta que el sueldo de un analista programador es de 23€/h y el de un director/a de 30€/h, se estiman los costes totales de la tabla 6.2. En esta tabla podemos ver la estimación de los costes humanos teniendo en cuenta las horas realizadas por ambos recursos.

Por último, en la tabla 6.3 podemos ver el coste total estimado para la realización de este proyecto, que asciende a 11 336,4€.

	Sueldo	Horas/Persona	Total
Directora de proyecto (x2)	30 €/h	21	1 260 €
Analista programador	23 €/h	431	9 913€
			11 173€

Tabla 6.2: Estimación del coste de los recursos humanos.

	Total
Recursos hardware	87,50 €
Recursos software	75.9 €
Recursos humanos	11 173 €
	11 336,4 €

Tabla 6.3: Estimación del coste total del proyecto.

Análisis de requisitos

EN este capítulo se analizan los objetivos que se pretende conseguir, se describen los actores que interactúan con la aplicación y se exponen las historias de usuario conseguidas a partir del análisis previamente realizado.

7.1 Requisitos funcionales

De acuerdo con los objetivos marcados al comienzo del proyecto, podemos extraer una serie de requisitos funcionales. Se ha decidido separarlos en dos grupos: el primero contiene los requisitos que puede llevar a cabo el usuario de tipo “Profesor”, que lo llamaremos profesor para simplificar, mientras que el otro grupo abarca los requisitos propios del usuario de tipo “Alumno”, que lo trataremos para simplificar como alumno.

7.1.1 Requisitos funcionales del profesor

- CU-01 Registro perfil profesor
- CU-02 Añadir alumno
- CU-03 Autenticación profesor
- CU-05 Cerrar sesión
- CU-06 Eliminar usuario
- CU-10 Configuración del profesor
- CU-12 Ver mis alumnos
- CU-13 Ver estadísticas
- CU-14 Importar alumnos

- CU-15 Exportar estadísticas
- CU-16 Generar correos

7.1.2 Requisitos funcionales del alumno

- CU-04 Autenticación alumno
- CU-05 Cerrar sesión
- CU-07 Completar ejercicios
- CU-08 Ver resultados
- CU-09 Realizar seguimiento
- CU-11 Configuración del alumno
- CU-13 Ver estadísticas

7.2 Requisitos no funcionales

La aplicación móvil no puede cerrarse inesperadamente ni aceptar valores incorrectos que lleven a un funcionamiento erróneo.

La interfaz debe ser clara, intuitiva y fácil de usar.

7.3 Actores

Se distinguen dos tipos de actores en función de su interacción con la aplicación. En la figura 7.1 puede verse la relación entre estos actores.

- **Usuario no autenticado:** Representa a los usuarios que no tienen cuenta o, en caso de tenerla, aún no han iniciado sesión con sus credenciales. Este tipo de usuarios solo tienen acceso al registro y autenticación de la aplicación.
- **Usuario autenticado:** Representa a los usuarios que se autenticaron con éxito proporcionando sus credenciales de autenticación (nombre de usuario y contraseña). El usuario autenticado puede pertenecer al perfil profesor o al perfil alumno. Dependiendo del tipo de perfil accederán a unas funcionalidades o a otras.

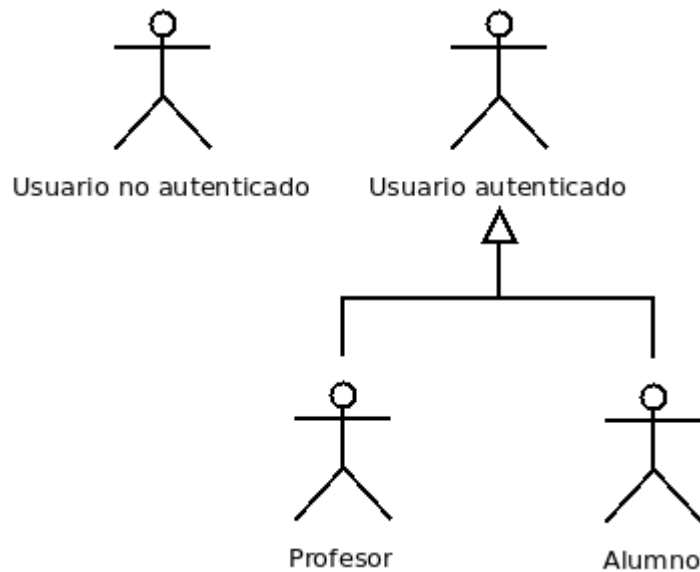


Figura 7.1: Diagrama de actores.

7.4 Términos de la aplicación

En esta parte se definen una serie de términos que se van a usar para explicar el análisis, diseño e implementación. Es fundamental entenderlos para comprender el flujo de determinadas partes de la aplicación.

- **Ejercicio:** Representa un cálculo, una comparación o cualquier prueba matemática que el alumno tiene que resolver.
- **Iteracion:** Se usa para representar una sucesión de ejercicios.
- **Nivel:** Cada nivel está formado por un conjunto de iteraciones, como mínimo una iteración y como máximo diez. Hay tres tipos de niveles en la aplicación: nivel 1, nivel 2 y nivel 3, siendo el nivel 1 el de menor dificultad y el nivel 3 el de mayor dificultad.
- **Bloque:** Comprende los distintos tipos de discalculia que se han implementado en la aplicación a partir de los tipos definidos por [3]: practognóstica, ideodiagnóstica, gráfica y operacional, que por simplicidad se llaman bloque 1, bloque 2, bloque 3 y bloque 4, respectivamente.
- **Curso:** Representa el curso en el que se encuentra el alumno. La aplicación dispone de cuatro cursos: 6º Infantil, 1º Primaria, 2º Primaria y 3º Primaria.

- **Seguimiento:** El seguimiento representa la situación actual del alumno dentro de la aplicación. Para ello se utilizan los siguientes datos: un campo curso para determinar el curso en el que se encuentra, un campo bloque para saber el tipo de discalculia que está practicando, un campo nivel para saber en que nivel se ubica actualmente, un campo número de iteración para guardar la iteración actual, un campo resultado para guardar el número de aciertos de la iteración, un campo tiempo para determinar la duración que le ha llevado completar la iteración y un campo fecha para almacenar el día que se completó la iteración.
- **Estadísticas:** Conjunto de datos originados a partir de la realización de iteraciones por parte de un alumno. Entre esos datos se encuentran: curso, bloque, nivel, iteración, número de aciertos, tiempo en completar la iteración y fecha en la que se realizó esa iteración.
- **Nombre de usuario:** Cadena de caracteres que identifica a los usuarios en la aplicación. Puede existir un alumno y un profesor con el mismo nombre de usuario pero no dos alumnos o dos profesores iguales.
- **Nombre del alumno:** Se ha decidido crear un campo para facilitarle a los profesores la identificación de sus alumnos. Será una decisión del profesor cómo usar este campo: nombre y apellidos del alumno, una descripción o lo que le resulte más fácil para distinguirlo. No es un valor obligatorio y pueden existir varios alumnos con el mismo nombre de alumno.
- **Mínimo de éxito:** Es un valor que determina si el alumno ha completado con éxito un nivel. El profesor decidirá, para cada alumno, qué valor es adecuado. Representa un porcentaje, de 10 en 10, desde el 10 % hasta un valor máximo del 100 %, lo que significa, respectivamente, que para la superación de un nivel se requiere responder acertadamente al 10 % de los ejercicios planteados y al total, es decir, al 100 % de ellos.

7.5 Historias de usuario

En este apartado se detallan las historias de usuario que se han obtenido a partir de los requisitos funcionales del sistema. Se han dividido en dos partes: la primera contiene las historias de usuario del profesor y la segunda las historias de usuario del alumno.

Por otro lado, a las tareas se les ha asignado una prioridad para decidir cuáles desarrollar primero. En este caso, una prioridad menor implica una mayor importancia en el proyecto. En la tabla 7.1 podemos ver las distintas prioridades que se han asignado a cada una de las tareas.

Nombre	Prioriad
CU-01 Registro perfil profesor	2
CU-02 Añadir alumno	2
CU-03 Autenticación profesor	2
CU-04 Autenticación alumno	2
CU-05 Cerrar sesión	3
CU-06 Eliminar usuario	2
CU-07 Completar ejercicios	1
CU-08 Ver resultados	1
CU-09 Realizar seguimiento	1
CU-10 Configuración del profesor	2
CU-11 Configuración del alumno	2
CU-12 Ver mis alumnos	2
CU-13 Ver estadísticas	2
CU-14 Importar alumnos	3
CU-15 Exportar estadísticas	3
CU-16 Generar correos	4

Tabla 7.1: Prioridades de los casos de uso.

7.5.1 Profesor

En este apartado se detallan las historias de usuario del profesor (ver figura 7.2).

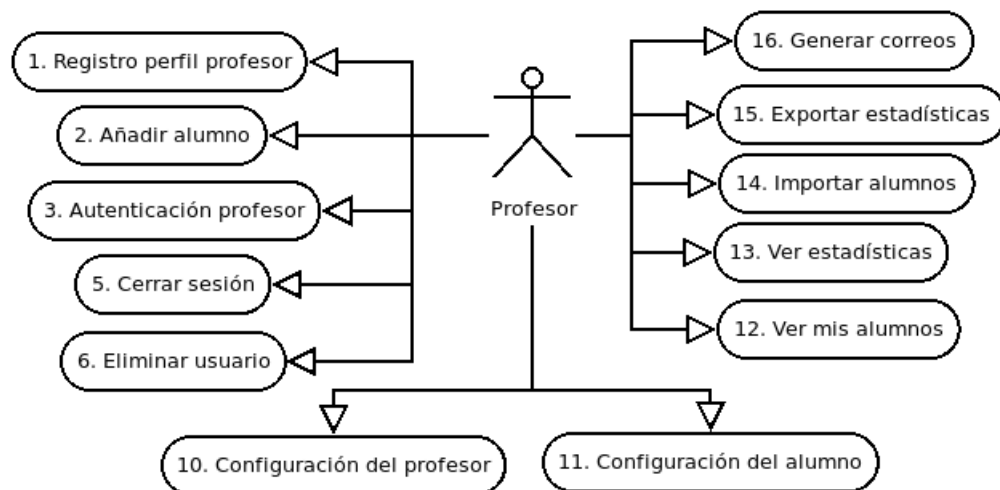


Figura 7.2: Historias de usuario del profesor.

CU-01 Registro perfil profesor: Para llevar a cabo el registro de un profesor en la aplicación, deberá introducir un e-mail (que no debe haber sido registrado previamente en la aplicación), una contraseña y una confirmación de la contraseña. El sistema comprobará que los datos introducidos son correctos y, en caso de serlo, creará un nuevo usuario de tipo “Pro-

fesor” y le enviará un correo de bienvenida. A continuación, el sistema le llevará a una pantalla donde podrá ver su perfil. En caso de que los datos introducidos sean incorrectos, se mostrará un mensaje indicando el error.

CU-02 Añadir alumno: Para llevar a cabo el registro de un alumno en la aplicación, el profesor deberá introducir los siguientes datos: nombre de usuario, nombre del alumno, contraseña, confirmación de la contraseña, curso, bloque, nivel, número de iteraciones y porcentaje mínimo de éxito. El sistema comprobará que los datos introducidos son correctos y, en caso de serlo, creará un nuevo usuario de tipo “Alumno”.

CU-03 Autenticación profesor: El profesor deberá introducir su nombre de usuario, contraseña y seleccionar el perfil “Profesor”. El sistema comprobará que las credenciales son correctas y le llevará a una pantalla donde podrá ver su perfil. En caso de que las credenciales sean incorrectas, se mostrará un mensaje indicando el error.

CU-05 Cerrar sesión: El usuario autenticado se convierte en usuario no autenticado. El sistema limpia sus datos de sesión y le lleva a la pantalla de inicio de sesión.

CU-06 Eliminar usuario: Partiendo del caso de uso CU-10 Configuración del profesor, el profesor puede eliminar las cuentas de sus alumnos y también su propia cuenta.

CU-10 Configuración del profesor: Partiendo del caso de uso CU-01 Registro perfil profesor o CU-03 Autenticación profesor, el sistema llevará al profesor a una pantalla donde puede ver y modificar sus datos.

CU-11 Configuración del alumno: Partiendo del caso de uso CU-12 Ver mis alumnos, el profesor también podrá acceder a la configuración de sus alumnos donde podrá ver y modificar los datos.

CU-12 Ver mis alumnos: El sistema mostrará al profesor un listado de sus alumnos divididos en cuatro grupos, uno para cada curso. Además del listado, esta pantalla contendrá diversos elementos que lanzarán otros casos de uso. Tendremos tres botones: el primer elemento permitirá importar múltiples alumnos mediante un archivo csv que lanzará el caso de uso CU-14 Importar alumnos; el segundo botón se usa para añadir un alumno, lo que iniciará el caso de uso CU-02 Añadir alumno, y el tercer elemento se usa para exportar las estadísticas de los alumnos seleccionados, lo que lanzará el caso de uso CU-15 Exportar estadísticas.

CU-13 Ver estadísticas: Partiendo del caso de uso CU-11 Configuración del alumno, el profesor podrá acceder al perfil de sus alumnos. Una vez dentro de la configuración de su alumno, al seleccionar el botón Ver estadísticas, el sistema le mostrará una nueva pantalla con las estadísticas de los niveles que ha realizado ese alumno. Los datos estadísticos que se muestran son los que se mencionan en la explicación anterior del término seguimiento.

CU-14 Importar alumnos: El profesor tiene la capacidad de añadir varios alumnos mediante la importación de un archivo en formato csv. Este archivo debe tener los siguientes campos: nombre de usuario, contraseña, nombre del alumno, curso, bloque, nivel, iteraciones y mínimo de éxito.

CU-15 Exportar estadísticas: El profesor tiene dos maneras de exportar las estadísticas de sus alumnos. La primera se encuentra al acceder a la configuración de un alumno, pero solo podrá exportar las estadísticas de ese alumno, mientras que la otra forma le permite exportar de forma conjunta las estadísticas de los alumnos que desee. En ambas situaciones, las estadísticas se exportarán a un archivo en formato csv utilizando el medio escogido por el profesor. Actualmente los medios que permiten una exportación correcta son: Drive, OneDrive, Yahoo Mail, Whatsapp, Telegram, Skype y Discord.

CU-16 Generar correos: Partiendo del caso de uso CU-01 Registro perfil profesor, se le enviará un correo de bienvenida al profesor en el que se le indicará su nombre de usuario. El profesor podrá solicitar una nueva contraseña si se olvida de la que tenía hasta ese momento. Recibirá un nuevo correo en el que se le indicará la nueva contraseña e instrucciones para cambiarla por otra.

7.5.2 Alumno

En este apartado se detallan las historias de usuario del alumno, mostradas en la figura 7.3.

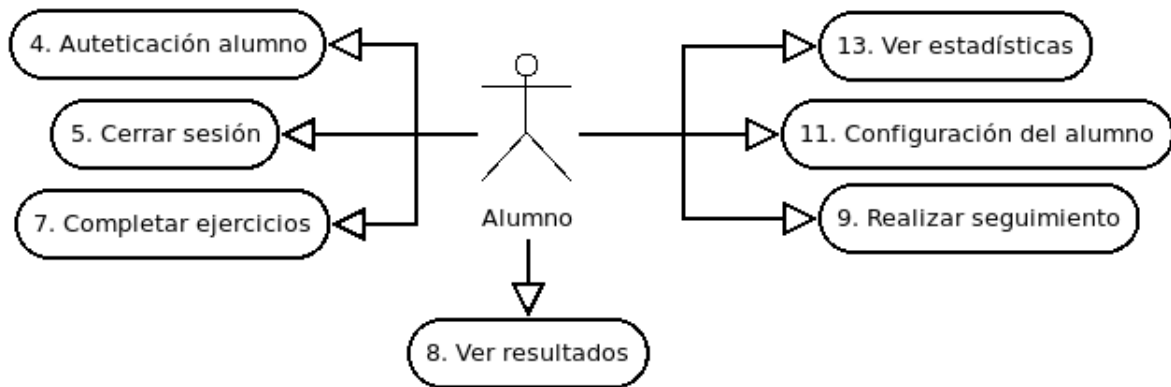


Figura 7.3: Historias de usuario del alumno.

CU-04 Autenticación alumno: El alumno deberá introducir su nombre de usuario, contraseña y seleccionar el perfil “Alumno”. El sistema comprobará que las credenciales sean correctas y le llevará a una pantalla donde seleccionará qué funcionalidad quiere realizar. En caso de que las credenciales sean incorrectas, se mostrará un mensaje indicando el error.

CU-05 Cerrar sesión: El usuario autenticado se convierte en usuario no autenticado. El sistema limpia sus datos de sesión y le lleva a la pantalla de inicio de sesión.

CU-07 Completar ejercicios: En función del curso, tipo de discalculia y nivel que el profesor ha determinado para el alumno, el sistema redirigirá a ese alumno a una pantalla donde realizará el nivel correspondiente dentro del bloque asignado. Cada nivel está estructurado en varias repeticiones de cinco ejercicios. Este número de iteraciones que realizará cada alumno es también ajustable por el profesor para cada uno de ellos. Una vez finalizada una iteración, se iniciará el caso de uso CU-10 Ver resultados.

CU-08 Ver resultados: El sistema lleva al alumno a una pantalla donde se muestran los resultados de la iteración y el tiempo que ha tardado en finalizar dicha iteración. El alumno decidirá si continua con la siguiente iteración, lo que le llevará al caso de uso CU-07 Completar ejercicios, o si prefiere, en cambio, tomarse un descanso. En el momento en el que el alumno termine su descanso, el sistema detectará cuál fue la última iteración completada para que pueda continuar el nivel en ese punto sin problemas.

CU-09 Realizar seguimiento: El sistema va guardando los datos de los ejercicios a medida que el alumno los va completando en cada iteración. Los datos que se guardan son los mencionados en la definición anterior del término seguimiento.

CU-11 Configuración del alumno: El sistema llevará al alumno a su perfil, donde podrá ver y modificar sus datos.

CU-13 Ver estadísticas: Partiendo del caso de uso CU-11 Configuración del alumno, el alumno podrá acceder a su perfil y, una vez ahí, seleccionando el botón Ver estadísticas, el sistema le mostrará una nueva pantalla con los resultados estadísticos de los niveles que ha completado. Estas estadísticas son las mismas que se han mencionado en el caso de uso CU-09 Realizar seguimiento.

Capítulo 8

Diseño

EN este capítulo se van a explicar las decisiones de diseño tomadas para el desarrollo de la aplicación.

8.1 Patrones de diseño

Se ha utilizado el patrón model-view-controller. Este patrón permite separar los componentes de la aplicación en función de la actividad para la que han sido creados. Separa los datos y la lógica de negocio de su representación gráfica y del componente que se encarga de controlar las eventos. Distinguimos tres partes o módulos:

- **Modelo:** Se encarga de guardar los datos de la aplicación y gestionar todos los accesos y modificaciones a esos datos. La base de datos que se ha utilizado forma parte del modelo.
- **Vista:** Representa la parte frontend de la aplicación y se encarga de la parte gráfica de la aplicación, le muestra al usuario final las pantallas, botones, imágenes, etc. En Android para la vista se utilizan unos archivos XML llamados layouts, en los que se definen los elementos visuales.
- **Controlador:** Representa la lógica de la aplicación y se encarga de gestionar todos los eventos recibidos. Hace la función de intermediario entre el modelo y la vista. Responde a los eventos que se originan en la vista y gestiona el flujo de datos entre el modelo y la vista. En el controlador se definen los casos de uso explicados en el capítulo anterior. Se implementarán en el lenguaje Java.

También se ha utilizado el patrón Singleton. Se encarga de garantizar que una clase solo tenga una instancia y proporcionar un punto global de acceso a ella. Este patrón se ha usado para la creación de la base de datos y así poder controlar y limitar la creación de sus instancias.

8.2 Diseño de la base de datos

Como podemos ver en la figura 8.1, tenemos siete tablas, aunque en este caso la dificultad no se encuentra en el número de tablas sino en las operaciones que se pueden hacer entre ellas.

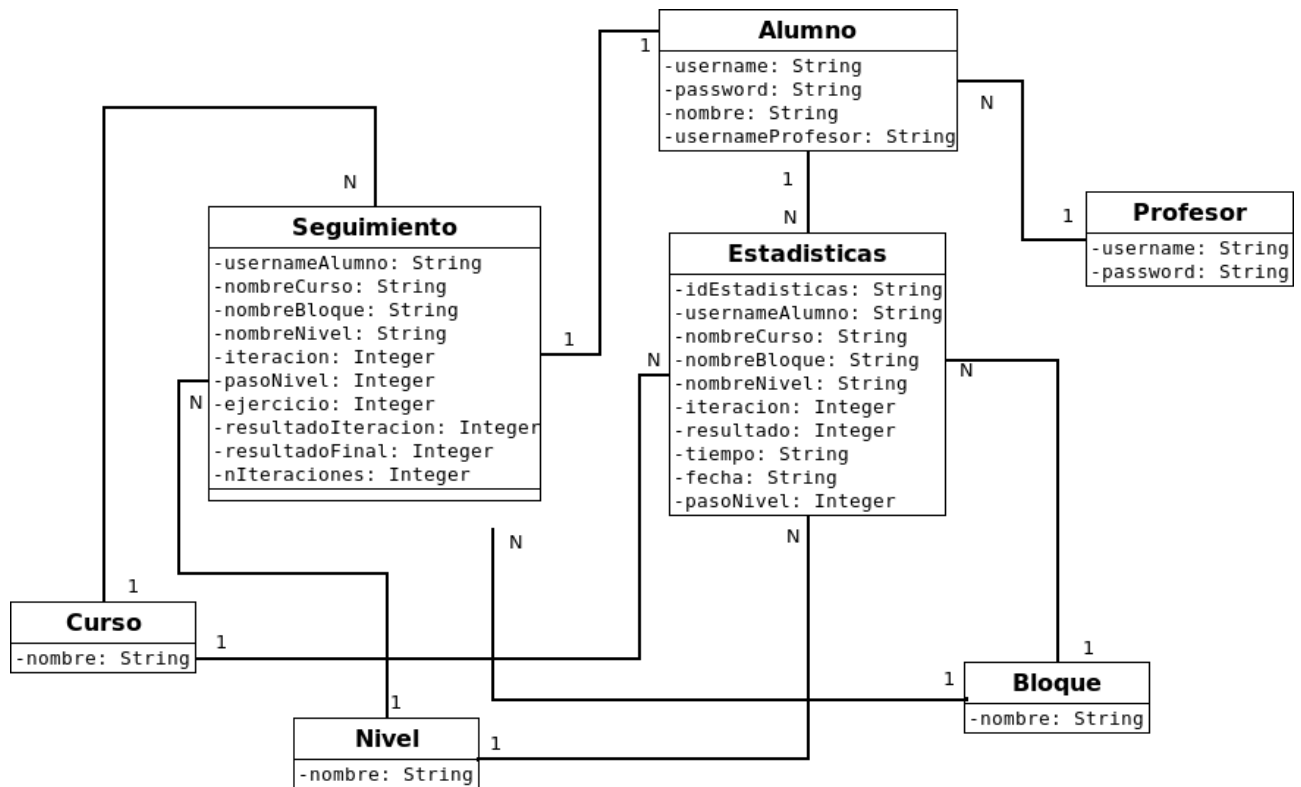


Figura 8.1: Modelo de datos.

A continuación explicaremos con más detalle cada una de estas tablas.

1. **Profesor**: Representa al usuario "Profesor". Se compone de nombre de usuario y contraseña. La clave primaria es el nombre de usuario, que será el id que use el profesor para autenticarse en la aplicación.
2. **Alumno**: Representa al usuario "Alumno". Sus columnas son nombre de usuario, contraseña, nombre del alumno y nombre de usuario del profesor. Tiene como clave primaria el nombre de usuario del alumno, que será el identificador que usará el alumno para autenticarse. Como clave foránea tenemos el nombre de usuario del profesor, lo que nos permitirá saber los alumnos que tiene cada profesor.
3. **Curso**: Almacena los distintos cursos que tenemos disponibles en la aplicación. Está formada únicamente por el nombre del curso que es la clave primaria.

4. **Bloque:** Almacena los distintos bloques con los que cuenta la aplicación. Está formada por el nombre del bloque, que es la clave primaria.
5. **Nivel:** Almacena los niveles de dificultad de la aplicación. Tiene un único campo que es el nombre del nivel y es la clave primaria.
6. **Seguimiento:** Se utiliza para poder tener un registro de la situación actual del alumno en la aplicación. Como clave primaria tenemos el nombre de usuario del alumno. En esta tabla tenemos tres claves foráneas: nombre del curso, nombre del bloque y nombre del nivel. Destacamos la columna iteración, que, junto con los nombres del curso, bloque y nivel, nos permiten saber qué ejercicio mostrar al alumno cuando éste se conecta a su sesión después de un tiempo de inactividad. Al igual que con la tabla Alumnos, tendremos una fila por cada alumno.
7. **Estadísticas:** Esta tabla almacena todos los datos que se generan cuando un alumno completa una iteración de un nivel. La clave primaria de esta tabla es un valor autoincremental.

8.3 Shared Preferences

Las bases de datos nos permiten almacenar grandes cantidades de información. Sin embargo, en ocasiones se necesita guardar un valor o pequeñas cantidades de información y el uso de una base de datos no sería eficiente. Por este motivo se ha decidido utilizar el API Shared Preferences, que nos permite almacenar pequeñas cantidades de información a través de un par clave-valor. Una vez que se genere una clave y se le asigna un valor, permanecerá en la aplicación hasta que se elimine esa clave, se borren todos los datos de la aplicación o se desinstale la misma.

8.4 Diseño de la aplicación

El diseño de la aplicación está formado por 16 actividades. A continuación explicaremos los detalles de estas actividades que forman los casos de uso. Para ello, al igual que en el análisis, las vamos a dividir en dos partes: la primera para el diseño de las tareas que tienen relación con el profesor y la segunda para las actividades del alumno.

8.4.1 Profesor

En esta parte explicaremos los casos de uso que tienen relación con el profesor (ver figura 7.2).

Registro y correo de bienvenida

Por simplicidad, tanto para alumnos como para profesores, se ha decidido que solo los profesores tienen la opción de registrarse en la aplicación. Se ha considerado que es la mejor solución, puesto que evita que los alumnos tengan que preocuparse por darse de alta. Cuando un profesor quiera registrarse, accederá a la pantalla de registro. Una vez que está en esa pantalla, introducirá un correo, una contraseña y la confirmación de ésta. Conviene indicar que se ha considerado que la contraseña deberá tener una longitud mínima de seis caracteres. Tras la introducción de los datos, el profesor pulsará el botón Aceptar, el sistema validará los datos y lanzará un Intent que le llevará a una nueva pantalla en la que podrá ver su perfil. Además, el profesor recibirá un mensaje en la dirección de correo electrónico introducido en el registro, en el que se le da la bienvenida a la aplicación y se le muestra el nombre de usuario que se le ha habilitado para usar en la misma, y que será su dirección de correo. Para el envío de correo se utiliza el API Javamail. La actividad Registro se encarga de realizar todo este proceso.

Ver mis alumnos y añadir uno nuevo

La actividad MisAlumnos es la encargada de mostrar el listado de los alumnos que tiene un profesor, mediante tablas con los nombres de los alumnos, una por cada curso. La tarea para añadir un alumno se lleva a cabo en la actividad RegistroAlumnos. Partiendo de la actividad anterior, MisAlumnos, el profesor pulsará el botón Añadir alumnos y el sistema lanzará un intent que le llevará a la nueva actividad RegistroAlumnos. Aquí se le habilitará un formulario para que el profesor rellene los datos del alumno que quiere dar de alta en la aplicación.

Autenticar profesor

El procedimiento de autenticación del profesor se lleva a cabo en la actividad Login. El profesor seleccionará el perfil Profesor e introducirá su nombre de usuario, su contraseña y pulsará el botón Iniciar Sesión. Una vez pulsado, el sistema validará los datos y lanzará un intent que llevará al profesor a una nueva pantalla en la que podrá ver sus datos. Si los datos introducidos no coinciden con ningún profesor registrado en el sistema se mostrará un mensaje de error.

Cerrar sesión

El proceso de cerrar la sesión de un profesor se realiza en la actividad SettingsProfesor. Esta actividad utiliza un fragment que se encarga de mostrar la configuración del profesor. Aquí se encuentra el botón de Cerrar sesión que, al pulsarlo, llevará al profesor a la pantalla

de inicio de sesión. Además de esta funcionalidad, este fragmento permite otras opciones de configuración que se explicarán más adelante.

Eliminar usuarios

El profesor tiene la capacidad de borrar del sistema las cuentas de sus alumnos y su propia cuenta. La eliminación de un alumno determinado se lleva a cabo en la actividad `SettingsAlumnoProfesor`. Esta actividad utiliza un fragment que es el que se encarga de la lógica de borrado de la cuenta del alumno escogido. Al eliminar su cuenta de profesor, se borrarán automáticamente todas las cuentas de sus alumnos. Esto se lleva a cabo en la actividad `SettingsProfesor`, que también utiliza un fragment que se encarga de la lógica de borrado del profesor y del borrado de todos sus alumnos.

Configuración del profesor

La configuración del profesor la realiza la actividad `SettingsProfesor` que, como se ha mencionado antes, utiliza un fragment que se encarga de la configuración del profesor. Esta tarea permite modificar y ver los datos del profesor, muestra su nombre de usuario, su contraseña, sus alumnos y le permite modificar también su contraseña como los ajustes de sus alumnos, que se explicarán en el siguiente caso.

Configuración del alumno

A pesar de que esta tarea es compartida por ambos usuarios, no la lleva a cabo la misma actividad dado que el funcionamiento es distinto en función del tipo de usuario que accede, es decir, no pueden realizar las mismas acciones un profesor o un alumno en la configuración. Aquí se va a explicar para el usuario profesor. El usuario alumno se encuentra en Configuración del alumno.

Un profesor, además de poder ver y modificar sus datos, también puede acceder a los datos de sus alumnos. La actividad que se ocupa de esta tarea es `SettingsAlumnoProfesor`, que utiliza un fragment que se encarga de la configuración del alumno. Este fragment le permite al profesor modificar y ver los datos del alumno seleccionado. Puede ver el nombre de usuario del alumno, sus estadísticas y puede además ver y modificar su contraseña, su nombre, su curso, el bloque, el nivel, el número de iteraciones y el mínimo de éxito exigido. Además, aquí se encuentra la funcionalidad de eliminar alumno, que ya se ha explicado previamente.

Ver estadísticas

Además de acceder a la configuración de sus alumnos, un profesor puede ver sus estadísticas. De esta tarea se encarga la actividad `DetallesAlumno`, que obtiene las estadísticas del

alumno seleccionado, genera una tabla con esos datos y los muestra en pantalla. Además, en esta actividad se encuentra también la funcionalidad de exportar las estadísticas de un alumno determinado.

Importar alumnos

En el capítulo 7.5 se ha visto que tenemos dos formas de dar de alta a un alumno en la aplicación, ambas realizadas por el profesor. La primera de ellas, que ya hemos explicado, consiste en dar de alta a un alumno a través de un formulario que será rellenado por el profesor. Esta forma de añadir un alumno tiene la limitación de que hay que rellenar un formulario por cada alumno que se quiera añadir. Por este motivo, se ha implementado otra alternativa que permita añadir varios alumnos simultáneamente y de forma automática, que se realizará mediante la importación de un fichero csv que contiene los datos de los alumnos. En la figura 8.2 se puede ver la secuencia que sigue el flujo de esta funcionalidad. Para utilizar esta segunda opción, partimos de la actividad MisAlumnos y el profesor pulsa la imagen de importación de csv (imagen con el nombre de csv y una flecha para arriba que indica que se está subiendo un archivo). Una vez pulsada, se le pedirá que acepte los permisos para acceder a la memoria del dispositivo y que seleccione el archivo csv que quiere importar. El sistema comprobará que el archivo tenga la estructura correcta, descartará los alumnos con nombres de usuario que ya existan en el sistema y dará de alta a los usuarios que han pasado estas validaciones. En la figura 8.3 se puede ver un ejemplo del archivo csv que se puede importar.

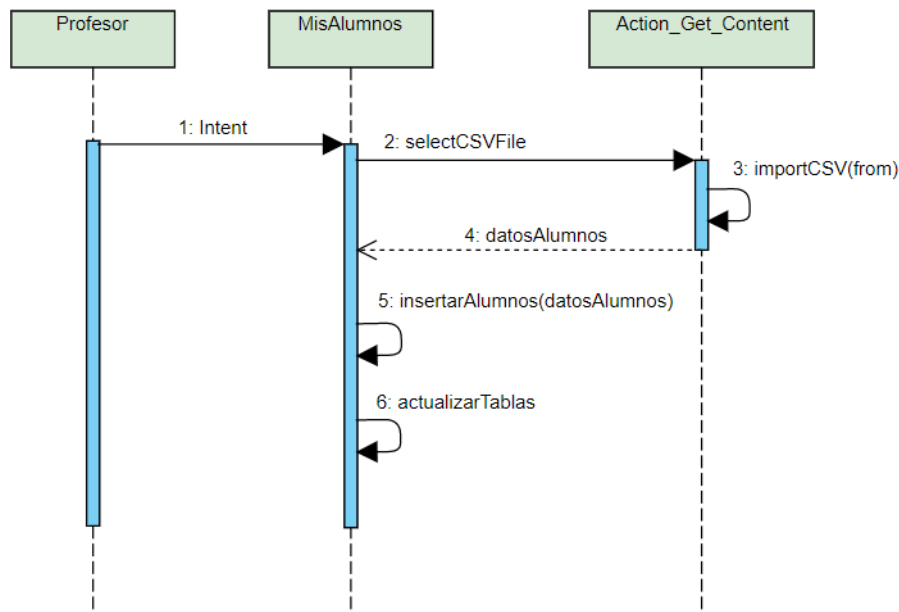


Figura 8.2: Diagrama de secuencia de Importar alumnos.

Nombre de usuario	Contraseña	Nombre del alumno	Curso	Bloque	Nivel	Iteraciones	Mínimo de éxito
david.d.f	david1	david diaz fernandez	Infantil	3	2	5	90
monica.f.b	monica1	monica diaz fernandez	1 Primaria	1	1	1	70
susana.g.a	susana1	susana diaz fernandez	Primaria 2	3	2	2	90
maria.h.f	maria1	maria diaz fernandez	Infantil	3	1	5	80
david.i.v	david2	david diaz fernandez	Primaria 1	2	3	10	90
alex.i.f	123456	alex diaz fernandez	Infantil	3	1	3	90
laura.k.x	laura4	laura diaz fernandez	Primaria 3	4	3	6	80
fran.l.f	123456	fran diaz fernandez	Infantil	1	1	5	90
juan.m.f	123456	juan diaz fernandez	Primaria 2	3	1	5	90

Figura 8.3: Ejemplo del archivo csv.

Exportar estadísticas

Hay dos formas de exportar las estadísticas. Hemos visto que la primera forma se encuentra en la actividad DetallesAlumno, en la que, además de mostrar la tabla con sus estadísticas, se da la posibilidad de exportarlas mediante un botón que pulsará el profesor. Esta primera forma es útil cuando se quieren exportar las estadísticas de un alumno determinado. Sin embargo, cuando la ocasión requiere exportar muchos alumnos, este proceso se hace ineficiente. Para solucionar esto, se ha desarrollado otra forma de exportación de los datos de los alumnos que permite seleccionar varios alumnos a la vez evitando tener que ir uno por uno. Para llevar a cabo esta funcionalidad, partimos de la actividad MisAlumnos, el profesor selecciona los alumnos que quiere exportar y pulsa sobre la imagen de exportación (imagen con el nombre de csv y una flecha hacia abajo para indicar que se exportan datos). El sistema lanzará un intent que le llevará a la actividad Exportar donde se le mostrarán los alumnos seleccionados. Al pulsar Aceptar se lanzará un Intent para que seleccione el medio por el cual realizar la exportación. En la figura 8.4 se puede ver la secuencia que sigue el flujo de esta funcionalidad. En ambas formas de exportación, la primera vez que el profesor exporte se le pedirá que acepte los permisos para acceder a la memoria del dispositivo. Es necesario que el profesor acepte estos permisos, de lo contrario no podrá llevar a cabo esta funcionalidad.

Generar correos

En el flujo de la aplicación existen dos situaciones en las que se envían correos. La primera situación se produce cuando un profesor se registra en la aplicación. Este caso se ha explicado ya con el registro en la aplicación. La otra situación surge cada vez que un profesor se olvida la contraseña y no puede acceder a la aplicación. Se ha decidido que, cuando esto ocurra, se genere y se envíe un correo al profesor indicando su nueva contraseña y los pasos a seguir para poder cambiarla en la aplicación. Este proceso se lleva a cabo en la actividad Login. Al seleccionar el perfil Profesor siempre se habilita una opción de restaurar la contraseña. Cuando el profesor seleccione esta opción, el sistema le pedirá el correo electrónico de registro para enviarle un mensaje a dicho correo para la restauración de contraseña.

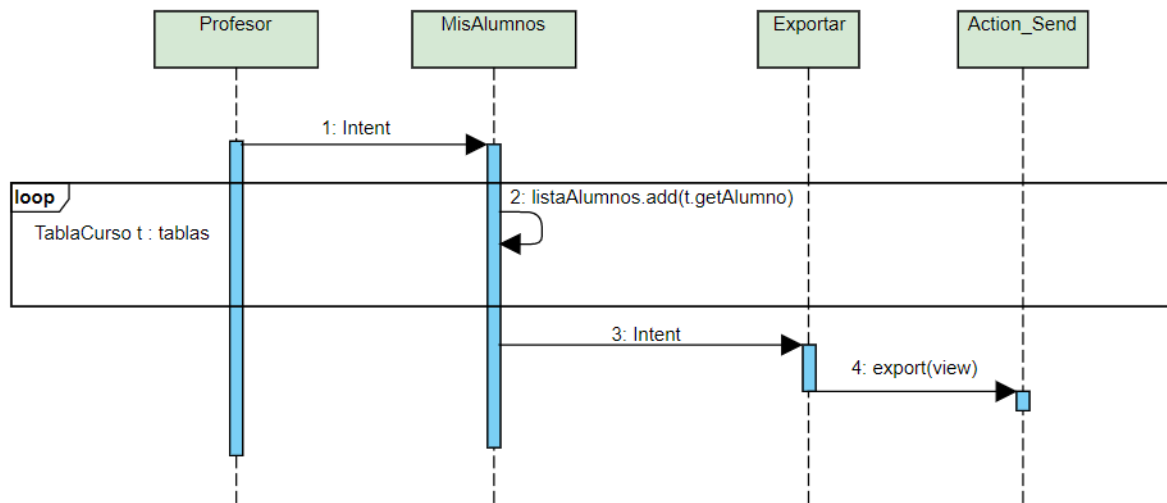


Figura 8.4: Diagrama de secuencia de Exportar estadísticas.

8.4.2 Alumno

Aquí se explican los casos de uso relacionados con el alumno (ver figura 7.3).

Autenticar alumno

El procedimiento de autenticación del alumno se lleva a cabo en la actividad Login. El alumno selecciona el perfil Alumno (por defecto, ya se encuentra seleccionado como Alumno) e introduce su nombre de usuario, su contraseña y pulsa el botón Iniciar sesión. Una vez pulsado, el sistema validará los datos y lanzará un intent hacia la actividad SeleccionarAccion donde seleccionará qué funcionalidad quiere realizar (acceder a los ejercicios, ver su perfil o cerrar sesión). A partir de ese momento ya se encuentra como usuario autenticado. Si los datos introducidos no coinciden con ningún alumno, se mostrará un mensaje de error.

Cerrar sesión

La tarea cerrar la sesión de un alumno se encuentra en dos lugares en la aplicación. Se ha decidido que exista más de una localización para que al alumno le sea fácil encontrar esta funcionalidad. Por una parte, tenemos la actividad SeleccionarAccion, que sería la pantalla principal del alumno, desde la que accede a todas sus funcionalidades. Al pulsar en el botón Cerrar sesión, se llevará a cabo esta funcionalidad. La otra forma de cerrar su sesión se encuentra en la actividad SettingsAlumno, que utiliza un fragment que se encarga de mostrar la configuración del alumno. Aquí se encuentra el botón de Cerrar sesión que, al pulsarlo, le

llevará a la pantalla de inicio de sesión. Además de esta funcionalidad, este fragmento permite otras opciones de configuración, que se explicarán más adelante.

Configuración del alumno

Las acciones que se permiten hacer en la configuración difieren en función del usuario que accede a esta tarea. De la configuración del alumno se encarga la actividad `SettingsAlumno`, que utiliza un fragment que se encarga de las acciones de la configuración del alumno. Se permite modificar y ver sus datos, se le muestra su nombre de usuario, contraseña, curso, bloque, nivel, sus estadísticas y se le permite modificar su contraseña.

Ver estadísticas

Partiendo de la actividad `SettingsAlumno`, un alumno puede acceder a sus estadísticas pulsando Ver estadísticas, que lanzará un intent a la actividad `PerfilAlumno`, la cuál genera una tabla con esa información y se la muestra por pantalla. A diferencia del profesor, el alumno no puede exportar sus estadísticas.

Completar ejercicios

Este caso de uso es el más grande en cuanto al número de clases que lo forman, ya que esta tarea se encarga de mostrar al alumno el ejercicio que le corresponde. La aplicación cuenta con 48 ejercicios en total, 12 por curso. Dentro de un mismo nivel, el ejercicio es el mismo pero variando de forma aleatoria los datos del mismo. Sin embargo, de un nivel a otro, aunque no se diferencian en la lógica que tienen, sí varía la dificultad. Por ejemplo, en los ejercicios de sumas, cuanto mayor es el nivel, es decir, la dificultad, mayores van a ser los números que se van a utilizar para realizar la operación. Sabiendo esto, se ha realizado un diseño escalable utilizando fragments. En total se han creado 34 fragments para los ejercicios. Se han agrupado los ejercicios similares bajo un mismo fragment que se encarga de mostrar los datos adecuados en función de la dificultad del ejercicio. Además de los fragments, tenemos la actividad `CompletarEjercicios`, que se encarga de crear el fragment correcto y mostrarlo. Este diseño hace posible la incorporación de futuros ejercicios e. incluso, el reemplazo de alguno de los actuales modificando muy poco el código. En la figura 8.5 se puede ver la secuencia que sigue el flujo de este apartado.

Ver resultados y realizar seguimiento

Partiendo de la actividad `CompletarEjercicio`, al terminar cinco ejercicios (una iteración), se lanza un intent hacia la actividad `MostrarResultadosIteracion` donde se muestra al alumno el resultado que ha conseguido, las preguntas que ha acertado y/o fallado y un logro en forma

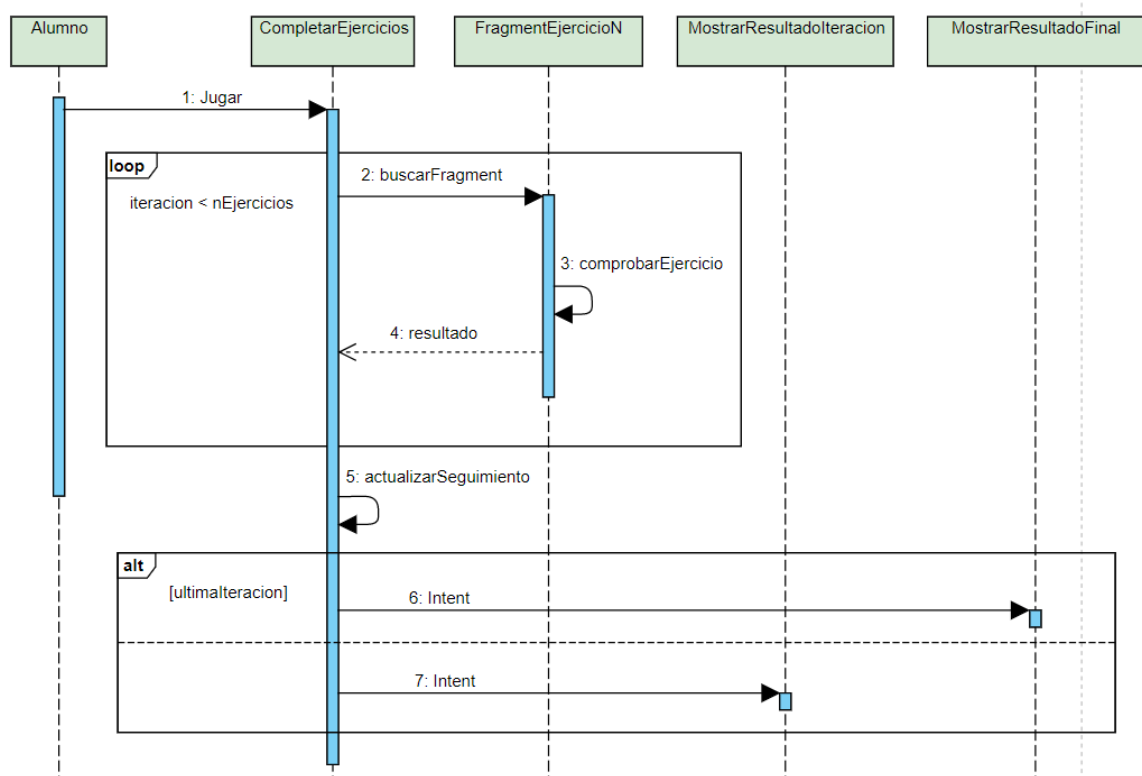


Figura 8.5: Diagrama de secuencia de Completar ejercicios.

visual para que el alumno no pierda la motivación y premiarle por completar ejercicios. Este logro será un caramelo cuando el alumno no complete con éxito la iteración, una medalla al completar con éxito una iteración y una copa al completar con éxito un bloque. Además, se actualiza su seguimiento con estos nuevos datos. Cuando el alumno completa la última iteración, se lanza un intent a la actividad `MostrarResultadoFinal` donde se le muestra al alumno el resultado de todas las iteraciones anteriores y, en caso de superar el porcentaje mínimo de éxito, se le permite avanzar al siguiente nivel. También se actualiza el seguimiento en esta actividad.

Implementación

EN este capítulo se comentan los aspectos más relevantes de la implementación de la aplicación.

9.1 Implementación

Librerías utilizadas

Se han usado librerías **Activation**, **Additionnal** y **Mail** para implementar la generación y el envío automático de correos que se llevan a cabo en la aplicación. Más adelante se mostrará un ejemplo de cómo se ha implementado esta generación de correos.

Se ha usado la librería **Picasso** para la carga automática de algunas imágenes que se emplean en la aplicación.

Se ha usado la librería **OpenCSV** para el tratamiento de archivos csv, tanto para importar como para exportar.

Por último, la librería **Preference** se ha utilizado para todo lo relacionado con la configuración del profesor y del alumno. Se ha usado para crear las actividades SettingsProfesor, SettingsProfesorAlumno, SettingsAlumno y sus respectivos fragments.

Estas librerías se añaden al gradle de nuestro proyecto en el apartado dependencies como se muestra en la siguiente imagen.

```
1 implementation files('libs/activation.jar')
2 implementation files('libs/additionnal')
3 implementation files('libs/mail.jar')
4 implementation 'com.squareup.picasso:picasso:2.71828'
5 implementation 'com.opencsv:opencsv:4.6'
6 implementation 'androidx.preference:preference:1.1.0'
```

Permisos Android

Para realizar determinadas acciones en Android se necesitan permisos especiales, como acceder a internet, a la cámara, al almacenamiento externo del dispositivo, etc. Los permisos que se vayan a utilizar se tienen que declarar en nuestro manifiesto, AndroidManifest. Este archivo contiene las configuraciones básicas de nuestra aplicación, como su versión e icono, la declaración de las actividades o los providers que se usan en la aplicación.

Nuestra aplicación requiere el permiso de acceso a internet para enviar los correos, el permiso de lectura del almacenamiento externo para importar los alumnos y el permiso de escritura del almacenamiento externo para exportar las estadísticas de los alumnos.

```

1 <uses-permission
    android:name="android.permission.ACCESS_NETWORK_STATE" />
2 <uses-permission android:name="android.permission.INTERNET" />
3 <uses-permission
    android:name="android.permission.READ_EXTERNAL_STORAGE" />
4 <uses-permission
    android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

```

Es necesario notificar a los usuarios de los permisos que se necesitan para la aplicación. Se deben comprobar que los permisos están aceptados y, en caso contrario, solicitárselos para conseguir que la aplicación funcione de manera adecuada.

```

1 private void permisosImportar () {
2     int permissionCheck = ContextCompat.checkSelfPermission(this,
3     Manifest.permission.READ_EXTERNAL_STORAGE);
4     if(permissionCheck!= PackageManager.PERMISSION_GRANTED){
5         ActivityCompat.requestPermissions(this, new
6         String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
7         MY_PERMISSIONS_REQUEST_READ_EXTERNAL_STORAGE);
8     } else {
9         selectCSVFile();
10    }
11 }

```

```

1 private void permisosExportar () {
2     int permissionCheck = ContextCompat.checkSelfPermission(this,
3     Manifest.permission.WRITE_EXTERNAL_STORAGE);
4     if(permissionCheck!= PackageManager.PERMISSION_GRANTED){
5         ActivityCompat.requestPermissions(this, new
6         String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE},
7         MY_PERMISSIONS_REQUEST_WRITE_EXTERNAL_STORAGE);
8     } else {
9         exportCSVFile();
10    }
11 }

```

```

1 @Override
2 public void onRequestPermissionsResult(int requestCode, String
3 permissions[], int[] grantResults){
4     if(grantResults.length > 0 && grantResults[0] ==
5     PackageManager.PERMISSION_GRANTED){
6         if (viewOpcion.getId() == R.id.exportarEstadisticas){
7             exportCSVFile();
8         } else {
9             selectCSVFile();
10        }
11    }
12 }

```

El método `onRequestPermissionsResult` es llamado la primera vez que el usuario acepta los permisos.

Implementación del registro del profesor

En el registro se comprueba que el profesor ha introducido datos válidos. Una vez hecho esto, se comprueba que el nombre de usuario del profesor no exista en la base de datos, concretamente en la tabla profesor. Después de comprobarlo, se empieza el proceso de envío de correo de bienvenida, que se explicará más adelante, y se realiza la inserción del nuevo profesor en la base de datos.

```
1 if (password1.isEmpty() || password2.isEmpty() ||
   username.isEmpty()) {
2     Toast.makeText(getApplicationContext(), "Rellene los campos
   obligatorios", Toast.LENGTH_SHORT).show();
3 } else if ((password1.length() < 6) && (password2.length() < 6)){
4     Toast.makeText(getApplicationContext(), "Las contraseñas
   deben tener más de 5 caracteres", Toast.LENGTH_SHORT).show();
5 } else if (!password1.equals(password2)) {
6     Toast.makeText(getApplicationContext(), "Las contraseñas
   deben ser iguales", Toast.LENGTH_SHORT).show();
7 } else if (!matcher.find()){
8     Toast.makeText(getApplicationContext(), "El nombre de
   usuario debe ser un correo electrónico",
   Toast.LENGTH_SHORT).show();
9 } else {
10     ArrayList <Profesor> lista = datos.obtenerProfesores();
11     profesorRepetido = false;
12     for (int i = 0; i < lista.size(); i++){
13         if (lista.get(i).getUsername().equals(username)){
14             Toast.makeText(getApplicationContext(), "Ya existe
   un usuario con ese nombre", Toast.LENGTH_SHORT).show();
15             profesorRepetido = true;
16             break;
17         }
18     }
19     if (!profesorRepetido){
20         //Enviamos el correo
21         ...
22         //Realizamos el proceso de dar de alta al profesor
23         String p = datos.insertarProfesor(new
   Profesor(username, password1));
24         editor.putString("profesor", p);
25         editor.commit();
26
27         Intent myIntent = new Intent(Registro.this,
   SettingsProfesor.class);
28         Registro.this.startActivity(myIntent);
29     }
30 }
31 }
32 }
```


Implementación del registro del alumno

Al igual que sucede con el profesor, es necesario comprobar que los datos son válidos y que el nombre de usuario del alumno está disponible. Una vez hecho eso, se empieza el proceso de inserción del alumno en la base de datos.

```

1  if (password1.isEmpty() || password2.isEmpty() ||
    username.isEmpty()) {
2      Toast.makeText(getApplicationContext(), "Rellene los campos
    obligatorios", Toast.LENGTH_SHORT).show();
3  } else if ((password1.length() < 6) && (password2.length() < 6)){
4      Toast.makeText(getApplicationContext(), "Las contraseñas
    deben tener más de 5 caracteres", Toast.LENGTH_SHORT).show();
5  } else if (!password1.equals(password2)) {
6      Toast.makeText(getApplicationContext(), "Las contraseñas
    deben ser iguales", Toast.LENGTH_SHORT).show();
7  } else {
8      ArrayList<Alumno> lista = datos.obtenerAlumnos();
9      alumnoRepetido = false;
10     for (int i = 0; i < lista.size(); i++){
11         if (lista.get(i).getUsername().equals(username)){
12             Toast.makeText(getApplicationContext(), "Ya existe
    un usuario con ese nombre", Toast.LENGTH_SHORT).show();
13             alumnoRepetido = true;
14             break;
15         }
16     }
17     if (!alumnoRepetido) {
18         String a = datos.insertarAlumno(new Alumno(username,
    password1,nombre,prefs.getString("profesor",null));
19         datos.insertarSeguimiento(new
    Seguimiento(a,curso,bloque,nivel, 0,pasoNivel, 0, 0, 0,
    nIteracion));
20         //Configurar sharedPreferences
21         ...
22         finish();
23     }
24 }

```

Implementación del login de los usuarios

En el siguiente código podemos ver cómo funciona el spinner que controla los dos tipos de usuarios: profesor y alumno. Cuando el perfil seleccionado en el spinner es profesor, habilitamos el enlace a la pantalla registro y el enlace de olvidar contraseña y, en caso contrario,

quitamos ambos enlaces.

```

1 cmbOpciones.setOnItemClickListener(
2     new AdapterView.OnItemClickListener() {
3         public void onItemClick(AdapterView<?>
4             parent, android.view.View v, int position, long id) {
5             opcion =
6             parent.getItemAtPosition(position).toString();
7             if (opcion.equals("Profesor")){
8                 setUsername.setHint("*Nombre de usuario
9                 (email)");
10                lblGotoContrasena.setVisibility(View.VISIBLE);
11                lblGotoRegister.setVisibility(View.VISIBLE);
12            } else {
13                setUsername.setHint("*Nombre de usuario");
14                lblGotoContrasena.setVisibility(View.GONE);
15                lblGotoRegister.setVisibility(View.GONE);
16            }
17        }
18    });

```

Implementación del flujo de llevar a cabo un ejercicio

El siguiente método onClick es llamado cada vez que el alumno presiona el botón siguiente en la pantalla CompletarEjercicios, que es la encargada de mostrar cada uno de ellos. Este método se encarga de ocultar el teclado y llamar al método comprobarEjercicio.

```

1 @Override
2 public void onClick(View v) {
3     InputMethodManager imm = (InputMethodManager)
4     getSystemService(Context.INPUT_METHOD_SERVICE);
5     imm.hideSoftInputFromWindow(v.getWindowToken(), 0);
6     click = prefs.getBoolean("click", false);
7     if (click) {
8         comprobarEjercicio();
9     }
10 }

```

Vamos a explicar el método comprobarEjercicio, que es llamado cada vez que el usuario pulsa el botón Siguiente. Este método se encarga de llevar a cabo una transición hacia al

siguiente ejercicio, o de mostrar el resultado de la iteración o resultado final en caso de que no hubiese más ejercicios. Se mostrará un resultado u otro en función de la iteración en la que se encuentre el alumno.

Se ha decidido utilizar manejadores (handlers) para conseguir una correcta transición hacia el siguiente ejercicio o resultado. Se han utilizado para que, en caso de que la transición no se completara, por ejemplo, si el alumno pone la aplicación en segundo plano o la cierra, se logre, en primer lugar, anular los cambios realizados y, en segundo lugar, evitar que la aplicación sufra errores inesperados. Debido a que el método comprobarEjercicio es grande, se ha decidido dividirlo en tres partes e ir explicando cada una de ellas.

En la primera parte empezamos configurando las variables y las sharedPreferences utilizadas para realizar la transición. Nos encontramos con el primer manejador, el cual se encarga de esconder el fragment que contiene el ejercicio y, por último, habilitamos y mostramos la imagen de acierto o fallo en función del resultado del ejercicio.

```

1 private void comprobarEjercicio() {
2     ejercicio++;
3     continuar = true;
4     enviar.setEnabled(false);
5     handler = new Handler();
6     //Escondemos el fragment que contiene el ejercicio
7     handler.postDelayed(new Runnable() {
8         public void run() {
9             continuar = true;
10            showHideFragment(fragment);
11            mostrarFragmento = 1;
12        }
13    }, 0);
14
15    //Habilitamos la imagen de la solución del ejercicio
16    imageView.setVisibility(View.VISIBLE);
17    imageView.animate().alpha(1).setDuration(1);
18
19    //Configuración de sharedPreference
20    editor.remove("click");
21    editor.remove("solucion1");
22    editor.remove("solucion2");
23    editor.remove("solucion3");
24    editor.commit();
25
26    //Configuramos la imagen en función de la solución del ejercicio
27    if (solucion.equals("Correcto") && (continuar)) {
28        imageView.setImageDrawable(getResources()
        .getDrawable(R.drawable.success_128));
    }

```

```

29         resultado = prefs.getInt("resultado", 0);
30         editor.putInt("resultado", ++resultado);
31         editor.putBoolean("Ejercicio" + ejercicio, true);
32         editor.commit();
33
34     } else if (solucion.equals("Incorrecto") && (continuar)){
35         imageView.setImageDrawable(getResources()
36             .getDrawable(R.drawable.fail_128));
37         editor.putBoolean("Ejercicio" + ejercicio, false);
38         editor.commit();
39     }

```

En la segunda parte continuamos la transición, en el if de la línea 1 entramos cuando quedan ejercicios por mostrar y el manejador se encarga de ocultar la imagen de acierto o fallo que mostramos en la primera parte, de deshabilitar el botón Siguiente y de mostrar el siguiente ejercicio. En el else se entra cuando el alumno se encuentra en el último ejercicio de una iteración. Aquí paramos el contador del tiempo total de la iteración y lo almacenamos usando las sharedpreferences. En el if de la línea 19 entramos cuando el alumno se encuentra en la última iteración del nivel, donde tenemos otro manejador que se encarga de actualizar el seguimiento en la base de datos, ocultar la imagen de acierto o fallo, mostrar el fragment que contiene el ejercicio y hacer un intent para mostrarle al alumno la pantalla de resultado final.

El manejador ocupa un tiempo de 800 ms. Explicaremos qué se ha escogido este valor y no otro cuando se concluye la exposición de las tres partes en las que se ha dividido este método.

```

1     if (ejercicio < nEjercicios) { //Mostramos el siguiente ejercicio
2         handler = new Handler();
3         handler.postDelayed(new Runnable() {
4             public void run() {
5                 imageView.setVisibility(View.INVISIBLE);
6                 showHideFragment(fragment);
7                 mostrarFragmento = 2;
8                 enviar.setEnabled(true);
9                 setUpFragment();
10            }
11        }, 800);
12    } else { //Último ejercicio de la iteración
13        final Context context = this;
14        tiempo.detener();
15        editor.putString("tiempo", tiempo.toString());
16        iteracion = seguimiento.getIteracion();
17        nIteraciones = seguimiento.getnIteraciones();
18    }

```

```

19         if (iteracion > nIteraciones - 1) { //Última iteración
20             handler = new Handler();
21             handler.postDelayed(new Runnable() {
22                 public void run() {
23                     //Calculamos el resultado final
24                     Seguimiento s =
25                     datos.obtenerSeguimientoByUsername(username);
26                     resultadoFinal = s.getResultadoFinal() +
27                     resultado;
28                     datos.actualizarSeguimientoByUsername(username,
29                     curso, bloque ,nivel, iteracion,pasoNivel, ejercicio, resultado,
30                     resultadoFinal,nIteraciones);
31
32                     editor.putInt("resultado_final",
33                     resultadoFinal);
34                     editor.commit();
35
36                     //Ocultamos la imagen de acierto o fallo y
37                     mostramos el fragmento que contiene el ejercicio
38                     imageView.setVisibility(View.INVISIBLE);
39                     showHideFragment(fragment);
40                     mostrarFragmento = 2;
41
42                     editor.remove("ejercicio");
43                     editor.remove("solucion");
44                     editor.commit();
45
46                     //Mostramos la pantalla de resultado final
47                     Intent myIntent = new Intent(context,
48                     MostrarResultadoFinal.class);
49                     startActivity(myIntent);
50                 }
51             }, 800);
52         } else { //Mostramos la siguiente iteración

```

En esta última parte se va a explicar lo que ocurre cuando el alumno no se encuentra en la última iteración del nivel, es decir, es la continuación del anterior else de la línea 45.

El código siguiente contiene otro manejador que se encarga de actualizar el seguimiento en la base de datos, ocultar la imagen de acierto o fallo, mostrar el fragment que contiene el ejercicio y hacer un intent para mostrarle al alumno la pantalla de resultado de la iteración. Al igual que el manejador de la segunda parte, ya explicada, tiene una duración de 800 ms. Se le ha asignado un valor distinto de cero para que la ocultación o aparición del fragment y de la imagen de acierto o fallo consiga un efecto de transición real.

```

1      handler = new Handler();
2      handler.postDelayed(new Runnable() {
3          public void run() {
4              //Vamos calculado el resultado final
5              Seguimiento s =
datos.obtenerSeguimientoByUsername(username);
6              resultadoFinal = s.getResultadoFinal() + resultado;
7              datos.actualizarSeguimientoByUsername(username,
curso, bloque,nivel, iteracion,pasoNivel, ejercicio, resultado,
resultadoFinal,nIteraciones);
8
9              editor.putInt("resultado_final", resultadoFinal);
10             editor.putInt("resultado_iteracion", resultado);
11             editor.commit();
12
13             imageView.setVisibility(View.INVISIBLE);
14             showHideFragment(fragment);
15             mostrarFragmento = 2;
16
17             editor.remove("ejercicio");
18             editor.remove("solucion");
19             editor.commit();
20
21             //Mostramos la pantalla del resultado de la
iteración
22             Intent myIntent = new Intent(context,
MostrarResultadoIteracion.class);
23             startActivity(myIntent);
24             }
25         }, 800);
26
27     }
28 }
29 }

```

Implementación del correo de bienvenida

En el código siguiente se muestra la lógica de la generación de correos de bienvenida. Para ello, primero configuramos una Session con autenticación SSL que se utilizará para el envío del correo, comprobamos si se ha creado la session, generamos el mensaje con el asunto, el destinatario, el cuerpo del mensaje, y lo enviamos.

```

1 AsyncTask.execute(new Runnable() {
2     @Override
3     public void run() {
4         StrictMode.ThreadPolicy policy = new
5         StrictMode.ThreadPolicy.Builder().permitAll().build();
6         StrictMode.setThreadPolicy(policy);
7
8         Properties properties = new Properties();
9         properties.put("mail.smtp.host", "smtp.gmail.com");
10        properties.put("mail.smtp.socketFactory.port", "465");
11        properties.put("mail.smtp.socketFactory.class",
12        "javax.net.ssl.SSLSocketFactory");
13        properties.put("mail.smtp.auth", "true");
14        properties.put("mail.smtp.port", "465");
15
16        try {
17            session = Session.getDefaultInstance(properties, new
18            Authenticator() {
19                @Override
20                protected PasswordAuthentication
21                getPasswordAuthentication() {
22                    return new
23                    PasswordAuthentication(correo,contrasena);
24                }
25            });
26            if (session!=null){
27                javax.mail.Message message = new
28                MimeMessage(session);
29                mensaje = "<h1>Te damos la bienvenida a
30                Discalculia.</h1>";
31                mensaje += "<h2>Con tu nueva cuenta, puedes
32                gestionar tus alumnos, " + "ver su seguimiento y exportar sus
33                datos.</h2>";
34                mensaje += "<h3>Usuario: " +
35                "<b>"+username+"</b>"+".</h3>";
36                message.setFrom(new InternetAddress(correo));
37                message.setSubject("Bienvenido a Discalculia");
38
39                message.setRecipients(javax.mail.Message.RecipientType.TO,
40                InternetAddress.parse(username));
41                message.setContent(mensaje,"text/html;
42                charset=utf-8");
43                Transport.send(message);
44                Toast.makeText(getApplicationContext(), "Correo
45                enviado correctamente", Toast.LENGTH_SHORT).show();

```

```

33         }
34     } catch (Exception e){
35         e.printStackTrace();
36     }
37 }
38 });

```

Implementación de exportar estadísticas

El procedimiento para exportar es el siguiente: creamos un archivo data.csv en el que insertamos las estadísticas de los alumnos, las cuales se encuentran almacenadas en la variable data (línea 4). Una vez hecho esto, vamos a utilizar un FileProvider para compartir el archivo con otras aplicaciones. Para ello, obtenemos el URI del archivo data.csv creado en la línea 1, después configuramos un intent que se encargará de mostrarle al profesor las aplicaciones con las cuales exportar los datos de sus alumnos. En el intent se añade el tipo del archivo (en nuestro caso text/csv), un nombre provisional para el archivo y la URI del archivo. Por último, ejecutamos el intent.

```

1 try{
2     //saving the file into device
3     FileOutputStream out = openFileOutput("data.csv",
4     Context.MODE_PRIVATE);
5     out.write((data.toString()).getBytes());
6     out.close();
7
8     //exporting
9     Context context = getApplicationContext();
10    File filelocation = new File(getFilesDir(), "data.csv");
11    Uri path = FileProvider.getUriForFile(context,
12    "com.example.myapplication.fileprovider", filelocation);
13    Intent fileIntent = new Intent(Intent.ACTION_SEND);
14    fileIntent.setType("text/csv");
15    fileIntent.putExtra(Intent.EXTRA_SUBJECT, "SeguimientoTotal");
16    fileIntent.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION);
17    fileIntent.putExtra(Intent.EXTRA_STREAM, path);
18    startActivity(Intent.createChooser(fileIntent, "Exportar"));
19 } catch (Exception e) {
20     e.printStackTrace();
21 }

```


Capítulo 10

Pruebas

EN este capítulo se comentan las pruebas realizadas. Durante el proceso de desarrollo se han llevado a cabo pruebas de caja negra para probar las distintas funcionalidades de la aplicación. También se han realizado pruebas de usabilidad por parte de las tutoras de este proyecto, lo que ha dado lugar a un mayor refinamiento de la aplicación. Se han realizado pruebas con los siguientes dispositivos: Samsung Galaxy S7 con Android 8, BQ Aquaris MS con Android 7.1.2, Samsung Galaxy J4+ con Android 9 y Samsung Galaxy Tab A T510 con Android 9.

10.1 Pruebas funcionales

Se han llevado a cabo pruebas funcionales para probar todas las funcionalidades de la aplicación. En este apartado comentaremos algunas de esas pruebas.

Generación de correos

Para probar el envío de correos, se da de alta un profesor y se comprueba que ha recibido el email de bienvenida. Para la prueba del correo de restauración de la contraseña se pulsa en el botón que dice “¿Te olvidaste de la contraseña?” de la pantalla de login, se introduce el correo de registro, después se accede al correo para ver la contraseña que nos ha generado la aplicación y se comprueba que nos deja entrar en la aplicación con la nueva contraseña.

Completar ejercicios

Se utiliza el profesor creado anteriormente para dar de alta a un alumno y probar que el flujo que se lleva a cabo a la hora de realizar los ejercicios es correcto. Se han probado todos los niveles de los cuatro bloques en cada uno de los cursos, es decir, un total de 48 niveles. Entramos con el nuevo alumno y realizamos las siguientes pruebas:

- Completamos el primer ejercicio y vemos que al pulsar el botón Siguiente nos muestra el siguiente ejercicio.
- Completamos una iteración que no sea la última, se muestra el resultado de la iteración y, al pulsar el botón Continuar, nos muestra el primer ejercicio de la siguiente iteración.
- Completamos la última iteración del nivel 1 con un acierto superior al mínimo de éxito del nivel, se muestra el resultado de la última iteración, al pulsar el botón Continuar muestra el resultado final y, al pulsar el botón Continuar, nos muestra el primer ejercicio del nivel 2.
- Completamos la última iteración del nivel 2 con un acierto superior al mínimo de éxito del nivel, se muestra el resultado de la última iteración, al pulsar el botón Continuar nos muestra el resultado final y, al pulsar el botón Continuar, nos muestra el primer ejercicio del nivel 3.
- Completamos la última iteración del nivel 3 con un acierto superior al mínimo de éxito del nivel, se muestra el resultado de la última iteración, al pulsar el botón Continuar nos muestra el resultado final y, al pulsar el botón Continuar, nos muestra la pantalla principal del alumno.
- Completamos la última iteración del nivel 1 con un acierto inferior al mínimo de éxito del nivel, se muestra el resultado de la última iteración, al pulsar el botón Continuar muestra el resultado final y, al pulsar el botón Continuar, nos muestra el primer ejercicio de la primera iteración del nivel 1.
- Completamos la última iteración del nivel 2 con un acierto inferior al mínimo de éxito del nivel, se muestra el resultado de la última iteración, al pulsar el botón Continuar nos muestra el resultado final y, al pulsar el botón Continuar, nos muestra el primer ejercicio de la primera iteración del nivel 2.
- Completamos la última iteración del nivel 3 con un acierto inferior al mínimo de éxito del nivel, se muestra el resultado de la última iteración, al pulsar el botón Continuar nos muestra el resultado final y, al pulsar el botón Continuar, nos muestra el primer ejercicio de la primera iteración del nivel 3.

Importar y exportar

Para probar que la importación funciona correctamente, pulsamos el icono importar, seleccionamos el archivo csv que queremos importar, finaliza el proceso de importar y nos muestra las tablas de los distintos cursos con los alumnos que se han podido insertar.

Para probar la exportación de un alumno, nos dirigimos como profesor a los datos de uno de nuestros alumnos, accedemos a sus estadísticas, pulsamos el icono de exportar y seleccionamos el medio por el cuál queremos exportar. Se comprueba que el archivo se ha exportado.

Para probar la exportación de varios alumnos, seleccionamos los alumnos que queremos exportar y pulsamos sobre el icono exportar, nos mostrará los alumnos que hemos seleccionado, pulsamos Aceptar y seleccionamos el medio por el cuál queremos exportar. Se comprueba que el archivo se ha exportado.

Configurar profesor y alumno

Utilizando un usuario "profesor" se ha probado la modificación de la contraseña en la configuración del profesor, también se han modificado en la configuración del alumno los datos relativos a un alumno, como su contraseña, nombre y los campos relacionados con su seguimiento, como curso, bloque, nivel, número de iteraciones o mínimo de éxito.

10.2 Pruebas de usabilidad

En este apartado vamos a comentar las distintas pruebas realizadas por los usuarios, en este caso las tutoras del proyecto.

En la siguiente lista se pueden ver las distintas pruebas de usabilidad realizadas:

- Alta de profesores y alumnos con formulario.
- Alta de alumnos con archivo csv.
- Prueba de todos los bloques y niveles.
- Pruebas de niveles cambiando el mínimo de acierto del alumno.
- Pruebas de exportación.
- Eliminación de usuarios.
- Utilización de la aplicación en distintos tipos de dispositivos.

Conclusiones

EN este proyecto se ha elaborado una aplicación que permite realizar intervenciones supervisadas por un tutor o profesor a niños/as que presentan discalculia o dificultades en el aprendizaje de las matemáticas. Este tipo de aplicaciones son importantes para conseguir una mejora en el aprendizaje, ya que contienen ejercicios pensados en la comprensión y no en la memorización y, además, se ajustan a la situación de cada alumno puesto que disponen de ejercicios con una dificultad adaptable por el profesor. El seguimiento del rendimiento académico y personal de cada alumno es posible, a partir de la resolución de ejercicios y juegos que les ayudan a entender y asimilar conceptos como la recta numérica, las figuras geométricas, la media, las relaciones de cantidad o el cálculo numérico, mediante la realización de sumas, restas, multiplicaciones y divisiones.

La tabla 11.1 muestra una comparación de nuestra aplicación con las distintas aplicaciones evaluadas en el capítulo 3. Si comparamos las cuatro aplicaciones, debemos decir que a nuestra aplicación le faltaría incorporar el auto-aprendizaje, ya que el alumno podría necesitar la ayuda del profesor sobre alguna cuestión matemática que no entiende. Por otra parte, se han implementado algunas funcionalidades que no tenían las aplicaciones ya existentes en el mercado y analizadas en ese capítulo, como son la exportación de información, en el caso de Smartick, la adaptación a distintos dispositivos, como es el caso de Dinamo Números, o el objetivo de centrarse con una profundidad mayor en la mejora del aprendizaje de las matemáticas en estudiantes con estas dificultades, lo que no sucede en el caso de Matific.

Tras la finalización de este Trabajo Fin de Grado, podemos decir que se han logrado todos los objetivos establecidos al inicio del mismo, esto es:

- Se ha logrado el objetivo principal de desarrollo de una aplicación para móviles y tablets Android que permite la intervención de la discalculia y ayude con el aprendizaje de las matemáticas a niños con esta dificultad.
- Se han incluido todos los cursos fijados al inicio: 6º de Educación Infantil y 1º, 2º y 3º

	Smartick	Dínamo Números	Matific	DiscalculiaTFG
Más de un tipo de dispositivo (móvil, tablet, web)	✓	X	✓	✓
Auto-aprendizaje	✓	✓	✓	X
Variedad de ejercicios	✓	✓	✓	✓
Controlar el seguimiento por parte de un profesor o tutor	✓	✓	✓	✓
Tiene como objetivo ayudar a mejorar la discalculia	✓	✓	X	✓
Exporta información del seguimiento del estudiante	X	✓	X	✓

Tabla 11.1: Comparación de la aplicación desarrollada con otras aplicaciones.

de Educación Primaria.

- La aplicación tiene dos perfiles de usuario distintos: profesor y alumno.
- Se ha cumplido con la exportación de información y se ha alcanzado un objetivo que no estaba inicialmente establecido: la importación de alumnos.
- Se han diseñado e implementado ejercicios/juegos matemáticos variados y con diferentes niveles de dificultad en función de la evolución de cada alumno.

Además, la realización de este Trabajo Fin de Grado ha permitido al alumno alcanzar la suficiente experiencia para llevar a cabo el análisis de requisitos de un proyecto, obtener los casos de uso, realizar el desarrollo de una aplicación móvil y realizar las pruebas necesarias.

11.1 Líneas futuras de trabajo

La aplicación está lista para ser probada en un entorno real y servir para el propósito por el que ha surgido este proyecto: intervenir en casos detectados de discalculia en un entorno educativo y conseguir mejoras en el aprendizaje de las matemáticas en estos casos.

Como trabajo futuro se podrían considerar los siguientes:

- La incorporación de más cursos (toda la Educación Primaria y los dos primeros cursos de la Educación Secundaria, al menos) para aumentar el rango de edades de los alumnos que podrían beneficiarse del uso de esta aplicación.
- Actualmente se exportan informes con las estadísticas de los alumnos en un formato csv. Sin embargo, se podría implementar la creación de ese mismo informe en formato pdf.

- La incorporación de una funcionalidad que permita reproducir audio en la aplicación, por ejemplo, que se muestre el enunciado de las preguntas utilizando mensajes de voz, dotar de voz a los logros del alumno, además del texto, etc.
- Tal y como hemos visto en el apartado anterior, la principal carencia de esta aplicación en comparación con otras es el auto-aprendizaje y la ayuda al alumno para que este pueda resolver un ejercicio ante el que no sabe cómo reaccionar.

Apéndices

Apéndice A

Planificación

En este apartado se incluyen los diagramas de Gantt de planificación del proyecto.

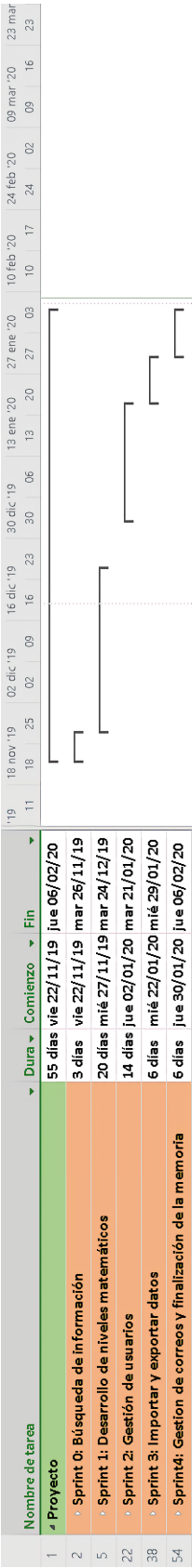


Figura A.1: Diagrama de Gantt de los Sprints del proyecto.

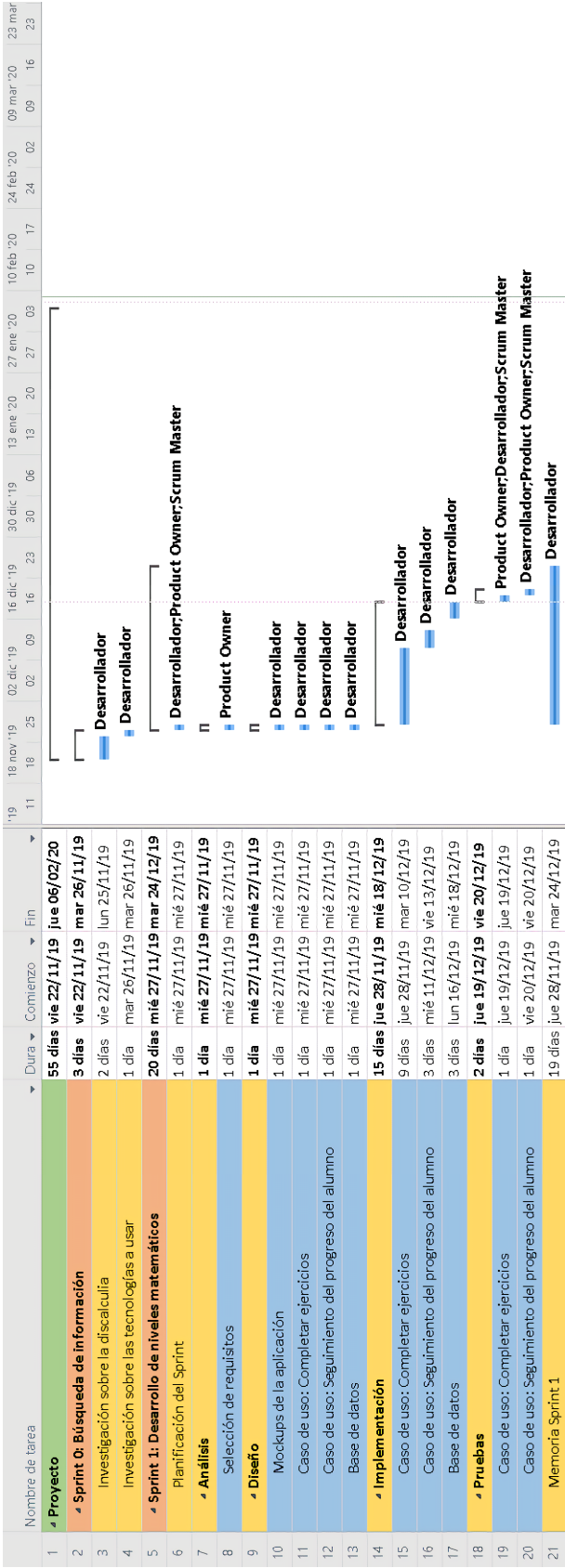


Figura A.2: Diagrama de Gantt de los Sprints 0 y 1.

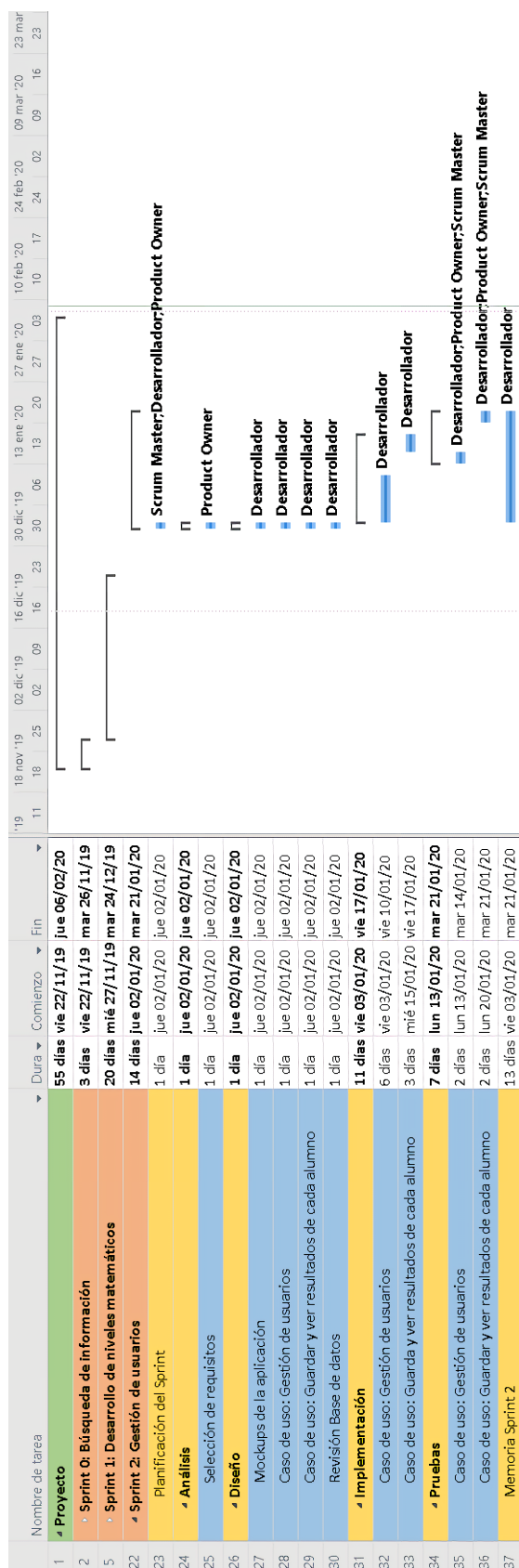


Figura A.3: Diagrama de Gantt del Sprint 2.

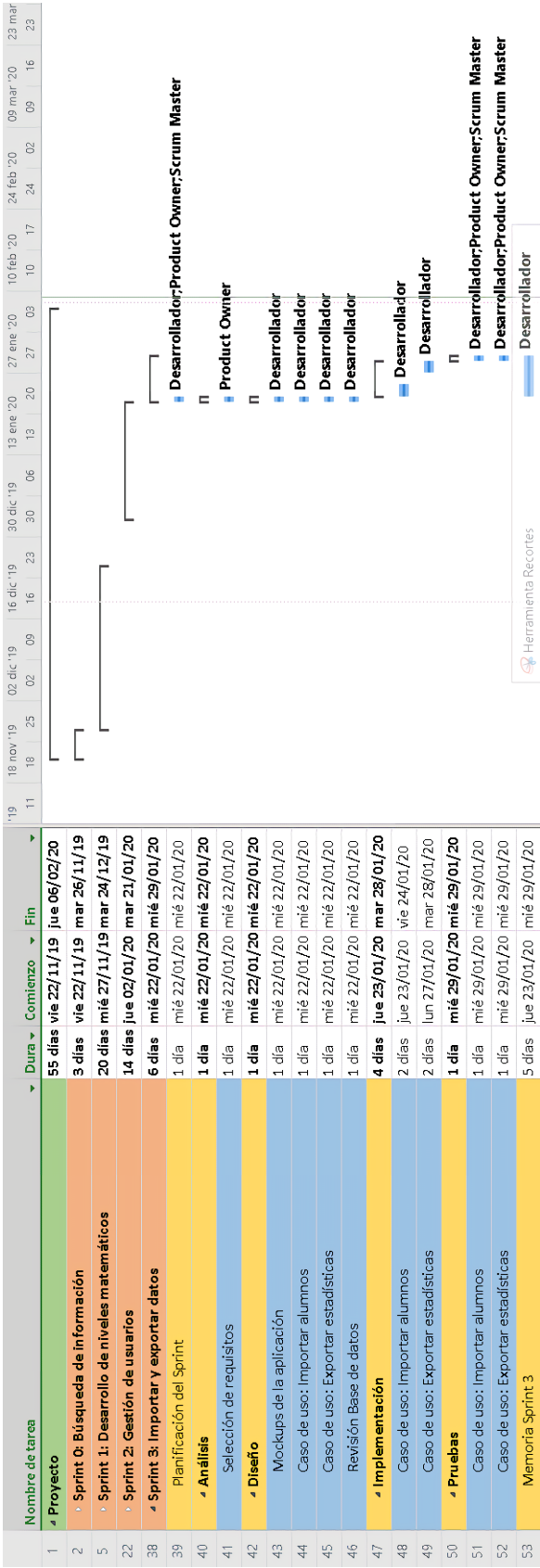


Figura A.4: Diagrama de Gantt del Sprint 3.

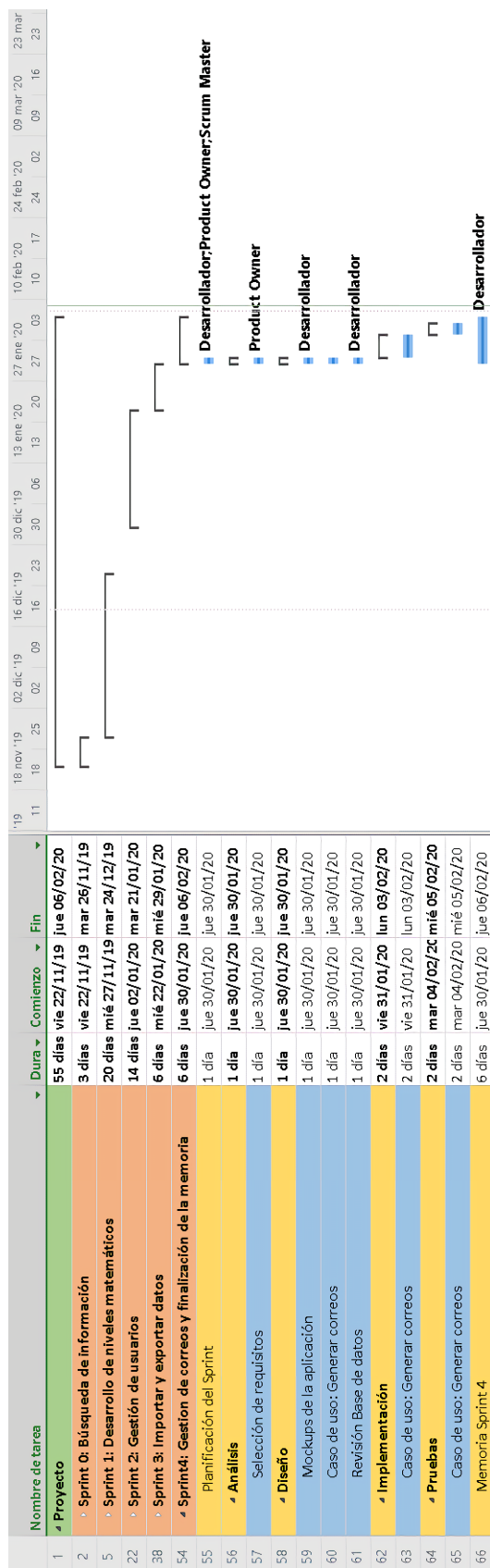


Figura A.5: Diagrama de Gantt del Sprint 4.



Figura A.6: Diagramas de las reuniones.



Figura A.7: Diagrama de seguimiento al 100%.

Competencias matemáticas

Las siguientes tablas recogen las competencias matemáticas establecidas en el DOG para los cursos de 6º de Educación Infantil y 1º, 2º y 3º de Educación Primaria.

En primer lugar, la tabla B.1 muestra las competencias matemáticas para el sexto curso de Educación Infantil (DOG Núm 121, Martes, 23 de junio de 2009, Pág 10.773 [32]).

Criterios de evaluación	Indicadores específicos
Exploración de los objetos y elementos del contorno inmediato y actuación sobre ellos. Reconocimiento de sus atributos y cualidades. Agrupación, clasificación y ordenación de estos elementos y colecciones, siguiendo distintos criterios.	Realización de series sencillas siguiendo un criterio dado. Ordenación de objetos siguiendo un criterio: tamaño, forma.
Utilización de los números para identificar, contar, clasificar, informarse y ordenar elementos de la realidad, aproximándose a su valor notacional y conceptual.	Utilización de los cuantificadores básicos (más, menos, igual). Conocimiento de la serie numérica. Utilización de la acción de contar como estrategia para obtener un dato numérico.
Propuesta y resolución de problemas simples. Reconocimiento de algunos aspectos geométricos básicos –líneas, puntos, rectángulos, cuadrados, triángulos, círculos, esferas, cubos.	Realización de cálculos mentales simples. Reconocimiento de algunas figuras geométricas: rectángulo, cuadrado, círculo, triángulo.

Tabla B.1: Competencias matemáticas de 6º de Educación Infantil.

La tabla B.2 muestra las competencias matemáticas para el curso de 1º de Educación Primaria (DOG Núm 154, Jueves, 14 de agosto de 2014, Pág 35042 [33]).

La tabla B.3 muestra las competencias matemáticas para el curso de 2º de Educación Primaria (DOG Núm 154, Jueves, 14 de agosto de 2014, Pág 35042 [33]).

Criterios de evaluación	Indicadores específicos
Iniciación en la utilización de los medios tecnológicos en el proceso de aprendizaje con la ayuda guiada del profesor o profesora.	Manifestación de interés en la utilización de los medios tecnológicos en el proceso de aprendizaje.
Lectura, escritura y ordenación de números enteros utilizando razonamientos apropiados.	Lectura, escritura y ordenación de números hasta el 99. El sistema de numeración decimal: valor de las posiciones de las cifras. Identificación del número mayor, el menor y el igual a uno dado.
Realización de operaciones y cálculos numéricos mediante diferentes procedimientos, incluido el cálculo mental, en situación de resolución de problemas.	Utilización de procedimientos diversos en la realización de cálculos numéricos básicos. Iniciación en el desarrollo de estrategias personales de cálculo mental. Gusto por la presentación ordenada y limpia de los cálculos y de los resultados. Cálculo de sumas y restas.

Tabla B.2: Competencias matemáticas de 1º de Educación Primaria.

La tabla B.4 muestra las competencias matemáticas para el curso de 3º de Educación Primaria (DOG Núm 154, Jueves, 14 de agosto de 2014, Pág 35042 [33]).

Criterios de evaluación	Indicadores específicos
Utilización, con la guía del profesor o profesora, de los medios tecnológicos en el proceso de aprendizaje y para la resolución de problemas.	Utilización de los medios tecnológicos en la resolución de problemas.
Lectura, escritura y ordenación de números enteros utilizando razonamientos apropiados.	Lectura, escritura y ordenación de números hasta el 999. El sistema de numeración decimal: valor de posición de las cifras. Identificación del número mayor, el menor y el igual a uno dado. Construcción de series ascendentes y descendentes.
Realización de operaciones y cálculos numéricos mediante diferentes procedimientos, incluido el cálculo mental, en situación de resolución de problemas. resultados. Cálculo de sumas. Cálculo de multiplicaciones. Resolución de operaciones con cálculo mental.	Iniciación en el desarrollo de estrategias personales de cálculo mental. Gusto por la presentación ordenada y limpia de los cálculos y de los
Discriminación de prismas, pirámides, conos, cilindros y esferas.	Identificación de cuerpos geométricos en objetos y ámbitos cotidianos: prisma, pirámide, cono, cilindro y esfera.

Tabla B.3: Competencias matemáticas de 2º de Educación Primaria.

Criterios de evaluación	Indicadores específicos
Lectura, escritura y ordenación utilizando razonamientos apropiados.	Lectura, escritura y ordenación de números hasta el 10 000. Identificación del número anterior y el siguiente a uno dado.
Interpretación de diferentes tipos de números según su valor.	Redondeo de números naturales.
Realización de operaciones y cálculos numéricos mediante diferentes procedimientos, incluido el cálculo mental, en situación de resolución de problemas. y división entera por un número de una cifra.	Realización de operaciones con números naturales: adición, sustracción, multiplicación

Tabla B.4: Competencias matemáticas de 3º de Educación Primaria.

Contenido del CD adjunto

Con este documento se adjunta un CD que contiene los siguientes archivos:

- **GuedeFernandez_Ruben_TFG_2020.pdf**. Documento en formato pdf con la memoria en formato electrónico.
- **GuedeFernandez_Ruben_TFG_2020_anexo.zip**: Fichero zip con la apk de la aplicación.

Manual de usuario. Aplicación móvil

La aplicación móvil desarrollada permite evaluar las competencias matemáticas mediante la realización de ejercicios y juegos encaminados a conseguir una mejora en su aprendizaje. El mínimo de API requerida para la aplicación es la 19, lo que quiere decir que la aplicación es compatible con el 95,3 % de los dispositivos móviles y tablets con sistema operativo Android.

D.1 Instalación de la aplicación

Para instalar la aplicación es necesario tener un dispositivo con una versión de Android 4.4 o superior, ya sea móvil o tablet.

El CD del proyecto contiene un archivo TFG.apk, que incluye los datos necesarios para instalar la aplicación.

Para esta instalación se debe permitir en el dispositivo (en ajustes del sistema) la instalación desde orígenes desconocidos.

A continuación se ejecuta el archivo desde el gestor de archivos del dispositivo. Una vez hecho esto, se arranca la aplicación (ver figura [D.1](#)).

D.2 Funcionamiento de la aplicación

Autenticación y registro

Al abrir la aplicación se muestra la actividad Login, con la cual se inicia sesión. El primer paso va a ser crear una cuenta. Para ello se selecciona el perfil profesor, que mostrará la pantalla de registro para darse de alta en el sistema (ver figura [D.2](#)).

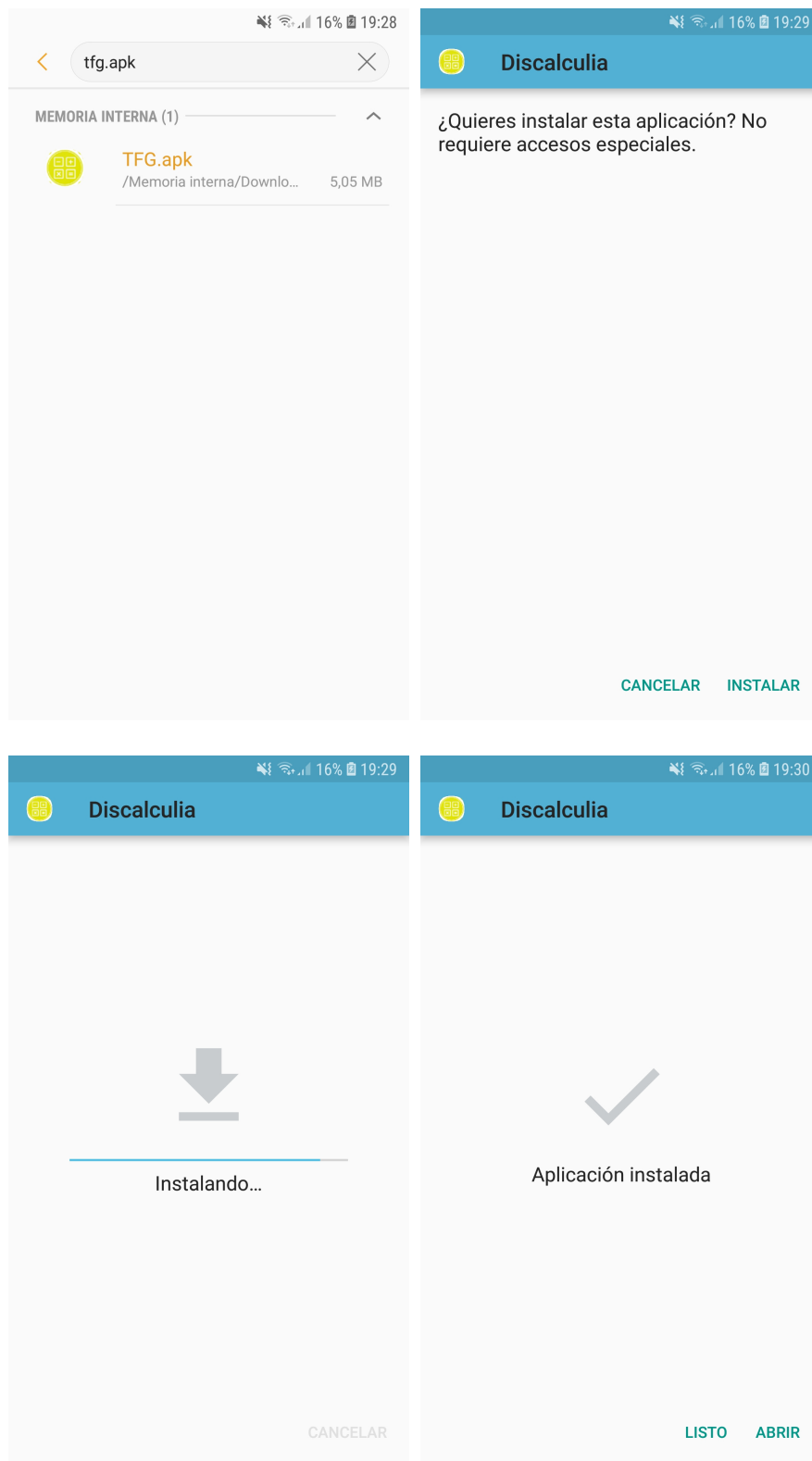


Figura D.1: Instalación de la aplicación.

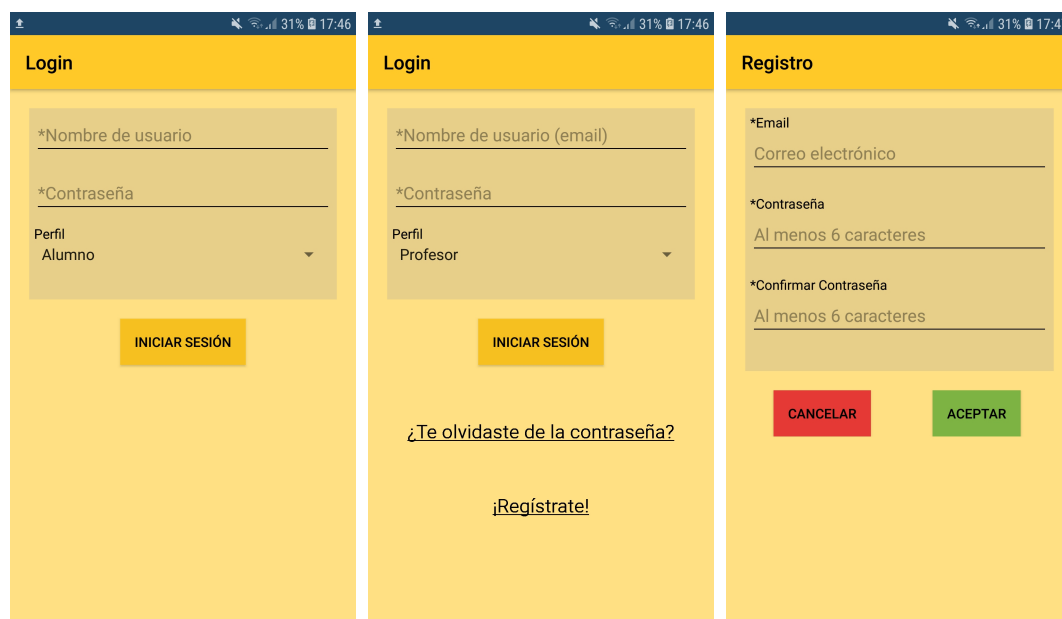


Figura D.2: Autenticación y registro.

Configuración del profesor

Una vez que estamos registrados, accedemos a la configuración del profesor: al pulsar en Contraseña, el profesor puede verla y cambiarla; pulsando en Eliminar mi cuenta se elimina la cuenta del profesor y las de todos sus alumnos, y para ver sus alumnos, se pulsa en Mis alumnos (ver figura D.3).

Importar uno o más alumnos

Para poder importar, el profesor tiene que estar en la pantalla MisAlumnos y pulsar en la imagen de la esquina inferior izquierda que pone csv con una flecha hacia arriba. Le pedirá que acepte los permisos y le mostrará una pantalla para seleccionar el archivo. Como resultado se mostrará la pantalla MisAlumnos con los alumnos importados (ver figura D.4).

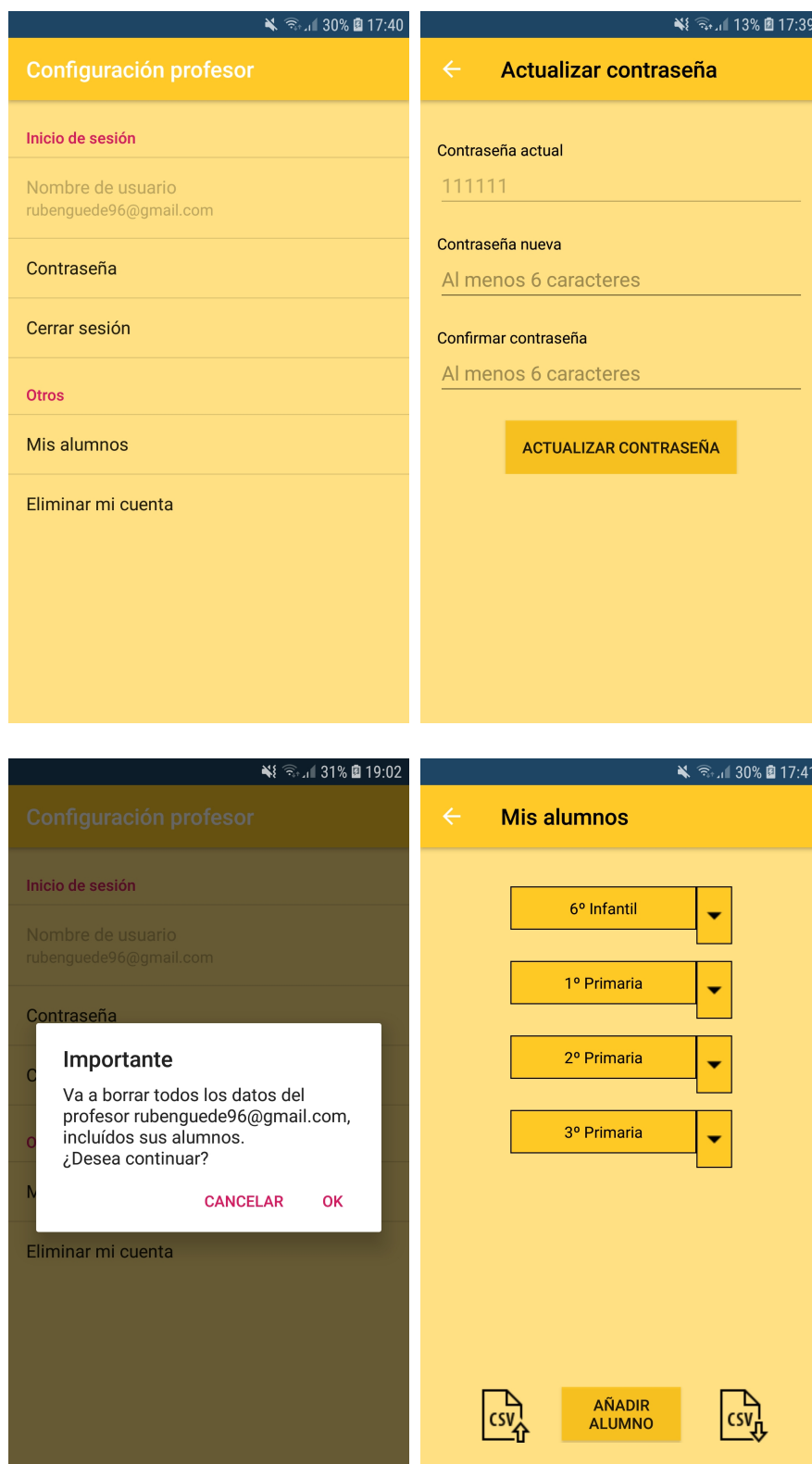


Figura D.3: Configurar profesor.

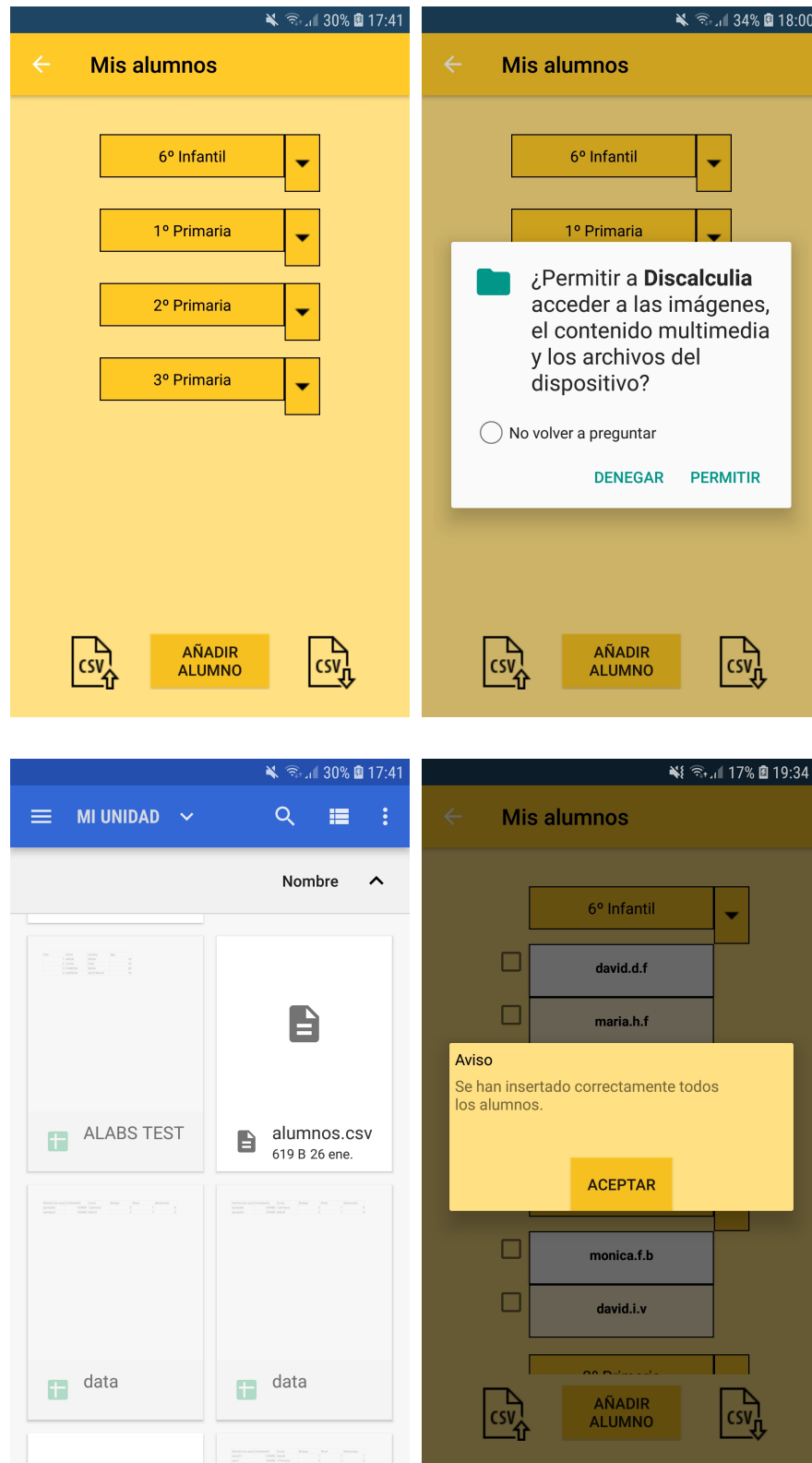


Figura D.4: Importar alumnos.

Añadir un alumno

Ahora vamos a ver la otra forma de añadir alumnos en la aplicación. Al pulsar el botón central Añadir alumno se muestra un formulario que hay que rellenar con los datos del alumno y pulsar el botón Aceptar. Como resultado se vuelve a la pantalla MisAlumnos donde se puede ver al nuevo alumno insertado (ver figura D.5).



Figura D.5: Añadir un alumno.

Configurar un alumno

En este apartado se explica cómo ver y configurar los datos de los alumnos de un profesor. Para ello, el profesor pulsa sobre un alumno dentro de la tabla y aparecerá la configuración de ese alumno. En esta nueva pantalla se puede ver su información, cambiar su contraseña, nombre de usuario y la configuración relacionada con su seguimiento: curso, tipo de discalculia, nivel, número de iteraciones por cada nivel y mínimo de éxito para dar por superado un nivel. El profesor puede eliminar la cuenta del alumno desde esta pantalla de configuración (ver figura D.6).



Figura D.6: Configurar Alumno.

Exportar estadísticas de un alumno

Para poder exportar las estadísticas de un alumno en concreto, el profesor pulsa Ver y exportar estadísticas desde la pantalla de configuración del alumno. Aquí el profesor pulsa la imagen de exportar situada debajo de la tabla. Le mostrará una pantalla para seleccionar el medio de exportación. Como resultado se exportarán las estadísticas y se mostrará de nuevo la tabla (ver figura D.7).

Exportar estadísticas de varios alumnos

Para poder exportar las estadísticas de varios alumnos, el profesor selecciona los alumnos que quiere exportar en la pantalla MisAlumnos. Una vez seleccionados, pulsa la imagen de exportar situada en la esquina inferior derecha. Le muestra una pantalla que contiene una tabla con los alumnos seleccionados y se le preguntará si quiere exportar sus estadísticas. Si acepta, le mostrará una pantalla para seleccionar el medio de exportación. Como resultado se exportaran las estadísticas y se mostrara de nuevo la lista de alumnos (ver figura D.8).

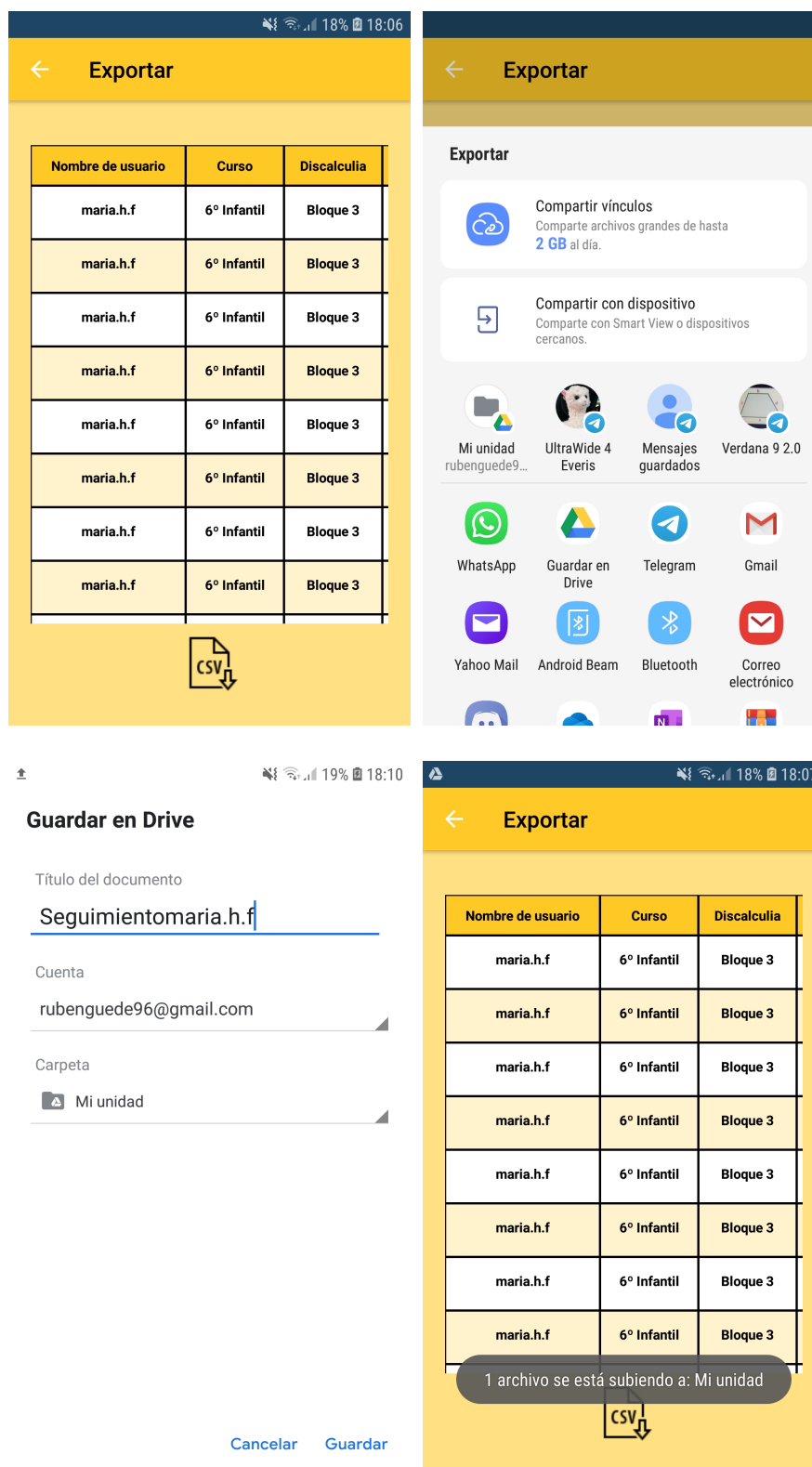


Figura D.7: Exportar estadísticas de un alumno.



Figura D.8: Exportar estadísticas varios alumnos.

Autenticación de un alumno

El alumno introduce sus credenciales y pulsa el botón Iniciar sesión. Se le muestra una nueva pantalla para elegir qué acción realizar (ver figura D.9).

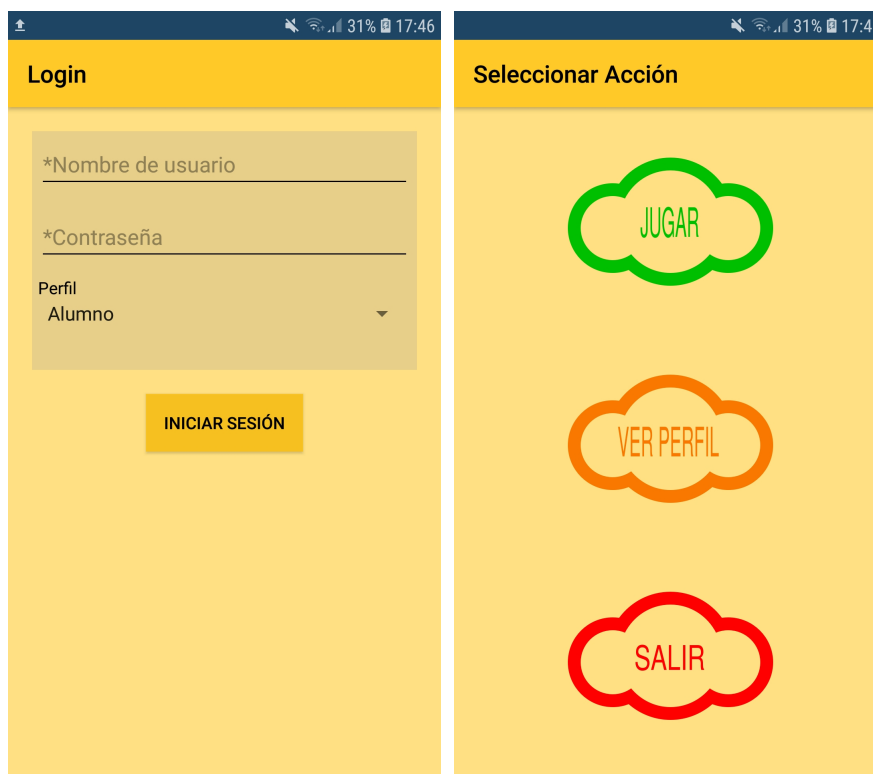


Figura D.9: Autenticar alumno.

Realizar ejercicios

Una vez que el alumno está autenticado, puede acceder a los ejercicios al pulsar en Jugar, y se le muestra el ejercicio correspondiente al curso, bloque y nivel que el profesor le ha asignado. Al completar cinco ejercicios se le muestran los resultados de esa iteración (ver figura D.10).

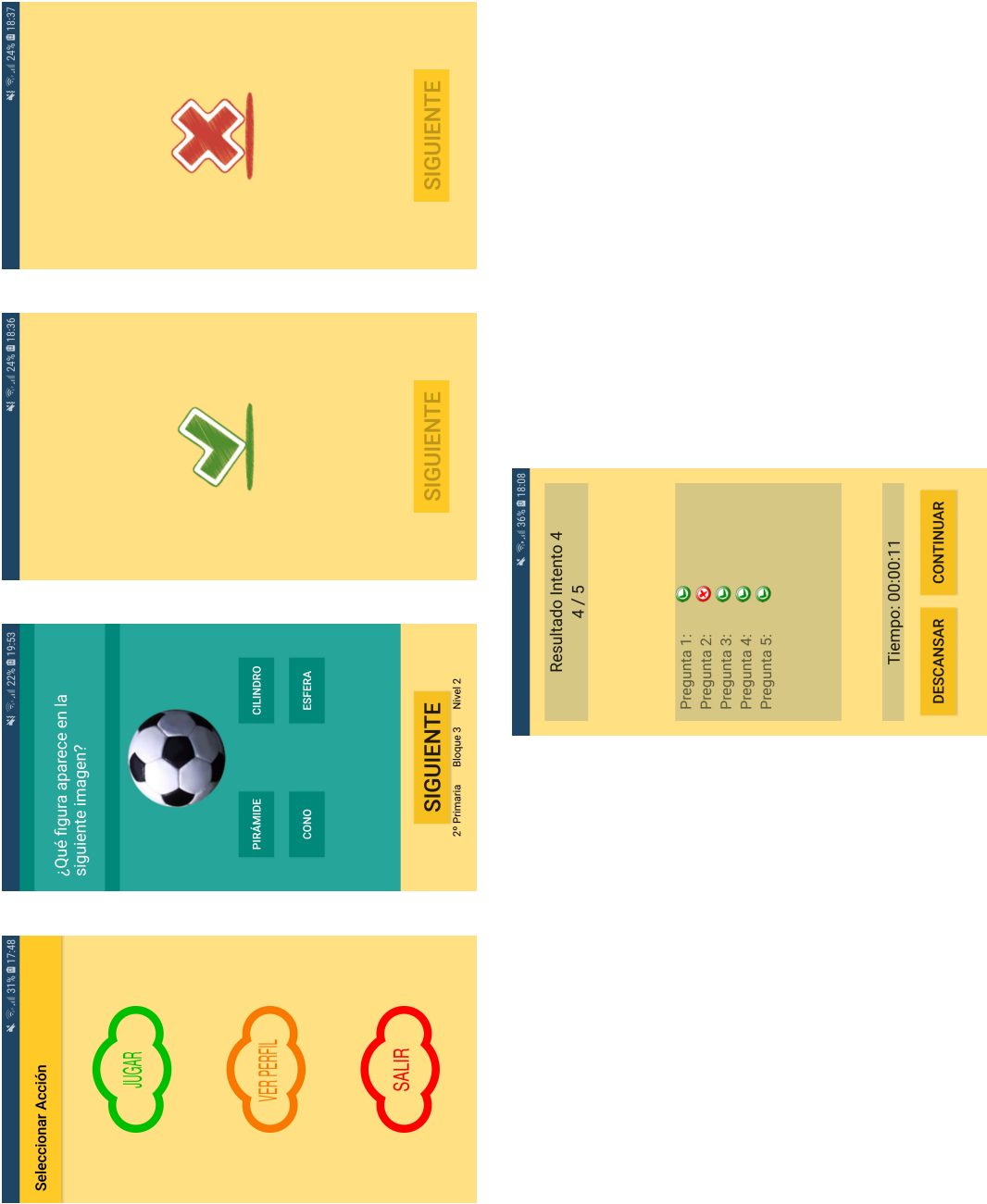


Figura D.10: Completar una iteración.

Completar un nivel

Cuando un alumno completa todas las iteraciones de un nivel, se le muestra el resultado final con un caramelo si no lo ha completado con éxito (esto es, no ha superado el mínimo porcentaje de éxito configurado por el profesor para este alumno) o una medalla si ha superado el nivel (es decir, ha superado el mínimo porcentaje de éxito exigido). Cuando completa el bloque se le muestra un trofeo (ver D.11).



Figura D.11: Completar un nivel.

Ver estadísticas

El alumno también puede ver sus estadísticas. Para ello, desde la pantalla principal en la que selecciona qué acción va a realizar, pulsa en Ver perfil. Aquí puede ver sus datos y al pulsar en Ver estadísticas se le muestra una tabla con esta información (ver figura D.12).



Figura D.12: Ver estadísticas.

Lista de acrónimos

- API** *Application Programming Interface..*
- BADYG** *Batería de Aptitudes Diferenciales y Generales.*
- BETT** *British Educational Training and Technology Show.*
- CI** *Coeficiente Intelectual.*
- CSV** *Comma-separated Values.*
- ERA** *Education Resources Awards.*
- SSL** *Secure Sockets Layer.*
- TDAH** *Trastorno por Déficit de Atención e Hiperactividad.*
- TEA** *Trastorno del Espectro Autista.*
- URI** *Uniform Resource Identifier.*
- WAIS** *Wechsler Adult Intelligence Scale.*
- WISC** *Wechsler Intelligence Scale for Children.*
- XML** *Extensible Markup Language.*

Glosario

Discalculia Dificultad en el aprendizaje de las matemáticas.

Discalculia gráfica Dificultad para escribir y usar símbolos matemáticos.

Discalculia ideognóstica Dificultad para entender conceptos matemáticos y para hacer cálculos mentales.

Discalculia operacional Falta de capacidad para realizar operaciones matemáticas.

Discalculia practognóstica Dificultad para enumerar, comparar y manipular objetos matemáticamente..

Fragment Representa un comportamiento o una parte de la interfaz de usuario.

Gradle Es un paquete de herramientas de compilación avanzadas, para automatizar y administrar el proceso de compilación.

Intent Es un objeto de mensajería que se puede usar para solicitar una acción de otro componente de una app.

Kinestésico Es un estilo de aprendizaje basado en la experimentación.

Startup Empresa que se encuentra en una edad temprana o de nueva creación y presenta grandes posibilidades de crecimiento.

Bibliografía

- [1] M. B. Teruel, N. Gil Novoa, A. I. Martín Ramos, and E. Vázquez Aguilar, “Medidas de reguerzo y apoyo educativos en la enseñanza obligatoria,” 2005. [En línea]. Disponible en: http://www2.escuelascaticas.es/publicaciones/GRATUITAS/Medidas_de_refuerzo.pdf
- [2] J. del Estado, “Ley orgánica 8/2013, de 9 de diciembre, para la mejora de la calidad educativa,” 2013. [En línea]. Disponible en: <https://www.boe.es/buscar/pdf/2013/BOE-A-2013-12886-consolidado.pdf>
- [3] J. Teruel Romero and A. Latorre Latorre, *Dificultades de aprendizaje*. Pirámide, 2014.
- [4] C. G. de Colegios Oficiales de Psicólogos, “Evaluación del test igf.” [En línea]. Disponible en: <https://www.cop.es/uploads/PDF/IGF.pdf>
- [5] T. K. N. Juan Antonio Amador Campos, Maria Forns Santacana, “La escala de inteligencia de wechsler para niños revisada.” [En línea]. Disponible en: <http://diposit.ub.edu/dspace/bitstream/2445/323/1/149.pdf>
- [6] C. G. de Colegios Oficiales de Psicólogos, “Evaluación del test wisc-iv.” [En línea]. Disponible en: <https://www.cop.es/uploads/PDF/WISC-IV.pdf>
- [7] N. M.-P. y V. N. C. Grégoire, J., “Tedi-math, test para el diagnóstico de las competencias básicas en matemáticas (2.^a ed) (manuel j. sueiro y jaime pereña, adaptadores).madrid: Tea ediciones,” 2015. [En línea]. Disponible en: http://www.web.teaediciones.com/Ejemplos/TEDI-MATH_Extracto_web.pdf
- [8] C. G. de Colegios Oficiales de Psicólogos, “Evaluación del cuestionario batería de aptitudes diferenciales y generales nivel m renovado,” 2015. [En línea]. Disponible en: <https://www.cop.es/uploads/PDF/2015/BADyG-CEPE.pdf>

- [9] Y. Campos, “Smartick deja atrás a métodos tradicionales como kumon,” 2017. [En línea]. Disponible en: <https://www.smartick.es/blog/noticias-smartick/smartick-en-los-medios/smartick-deja-atras-kumon/>
- [10] A. A. of Pediatrics, “Tipos de problemas de aprendizaje,” 2017. [En línea]. Disponible en: <https://www.healthychildren.org/Spanish/health-issues/conditions/learning-disabilities/Paginas/types-of-learning-problems.aspx>
- [11] G. de Navarra Departamento de Educación, “Definición: La discapacidad motora,” 2017. [En línea]. Disponible en: https://drive.google.com/file/d/1hoQbM4O04k6IZAOU7qtrxh9n_3vXoyJQ/view
- [12] E. Occidental, “Síndrome de down. principales características,” 2017. [En línea]. Disponible en: <https://www.espaciologopedico.com/noticias/det/7471/sindrome-de-down-principales-caracteristicas.html>
- [13] R. Lord, “Síndrome de asperger en la infancia.” [En línea]. Disponible en: https://www.asperger.es/sindrome_asperger_infancia_sintomas.html
- [14] J. P. Ltd, “Dinamo números.” [En línea]. Disponible en: <https://dinamonumeros.com/dynamo-maths/>
- [15] [En línea]. Disponible en: <http://bettawards.com/about-bett-awards/>
- [16] [En línea]. Disponible en: http://www.educationresourcesawards.co.uk/about_awards.htm
- [17] Matific, “Matific.” [En línea]. Disponible en: <https://www.matific.com/es/es/home/our-product/our-content/>
- [18] “Microsoft project.” [En línea]. Disponible en: <https://products.office.com/es-es/project/project-management-software>
- [19] Git. [En línea]. Disponible en: <https://git-scm.com/>
- [20] I. GitHub. [En línea]. Disponible en: <https://github.com/>
- [21] GNU. [En línea]. Disponible en: <https://wiki.gnome.org/Apps/Dia/>
- [22] V. Paradigm. [En línea]. Disponible en: <https://online.visual-paradigm.com/es/>
- [23] [En línea]. Disponible en: <https://developer.android.com/studio>

- [24] G. Castro, “Sqlite: Sistema de gestión de datos relacional embebidos.” [En línea]. Disponible en: <https://javanicaragua.org/2019/08/03/sqlite-sistema-de-gestion-de-datos-relacional-embebida/>
- [25] Mozilla and individual contributors, “Xml: Extensible markup language.” [En línea]. Disponible en: <https://developer.mozilla.org/en-US/docs/Web/XML>
- [26] [En línea]. Disponible en: <https://www.sqlite.org/index.html>
- [27] U. C. los Ángeles Chimbote, “Metodología de desarrollo software,” 2017. [En línea]. Disponible en: <https://www.uladech.edu.pe/images/stories/universidad/documentos/2018/metodologia-desarrollo-software-v001.pdf>
- [28] [En línea]. Disponible en: <https://www.atlassian.com/es/software/jira>
- [29] [En línea]. Disponible en: <https://www.collab.net/products/versionone>
- [30] [En línea]. Disponible en: <https://trello.com/es>
- [31] K. Schwaber and J. Sutherland, “La guía de scrum,” 2016. [En línea]. Disponible en: <https://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Spanish.pdf#zoom=100>
- [32] [En línea]. Disponible en: https://www.xunta.gal/dog/Publicados/2009/20090623/Anuncio24C76_gl.html
- [33] [En línea]. Disponible en: https://www.xunta.gal/dog/Publicados/2014/20140814/AnuncioG0164-310714-0001_gl.html

